

# vSphere IaaS 制御プレーン のバックアップとリストア

Update 3

VMware vSphere 8.0

VMware vCenter 8.0

VMware ESXi 8.0

VMware by Broadcom の Web サイトで最新の技術ドキュメントを確認できます

<https://docs.vmware.com/jp/>

**VMware by Broadcom**

3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2024 Broadcom. All Rights Reserved. 「Broadcom」という語表現は、Broadcom Inc. およびその子会社のいずれかまたは両方を指します。詳細については、<https://www.broadcom.com> を参照してください。本書に記載されるすべての商標、製品名、サービス マークおよびロゴは、各社に帰属します。

# 目次

- vSphere IaaS 制御プレーンのバックアップとリストア 4
- 1 vSphere IaaS Control Plane のバックアップとリストアに関する考慮事項 5**
- 2 スーパーバイザー 制御プレーンのバックアップとリストア 7**
  - スーパーバイザー の状態のバックアップ 7
  - スーパーバイザー 制御プレーンのリストア 9
- 3 スーパーバイザー での Velero Plugin for vSphere のインストールと構成 10**
- 4 TKG サービス クラスタとワークロードのバックアップとリストア 21**
  - TKG サービス クラスタとワークロードのバックアップとリストアに関する考慮事項 21
  - Velero Plugin for vSphere を使用した TKG クラスタ ワークロードのバックアップとリストア 22
    - TKG クラスタでの Velero Plugin for vSphere のインストールと構成 22
    - Velero Plugin for vSphere を使用した TKG クラスタ ワークロードのバックアップとリストア 27
  - スタンドアロンの Velero と Restic を使用した スーパーバイザー での TKG クラスタ ワークロードのバックアップとリストア 29
    - TKG クラスタへのスタンドアロン Velero と Restic のインストールと構成 29
    - スタンドアロンの Velero と Restic を使用したクラスタ ワークロードのバックアップとリストア 34
  - Velero と CSI スナップショットを使用したバックアップとリストア 42
- 5 vSphere IaaS Control Plane での仮想マシン サービスの仮想マシンのバックアップとリストア 45**
  - 仮想マシン サービスの仮想マシンの手動登録 47
- 6 Velero Plugin for vSphere を使用した vSphere ポッド のバックアップとリストア 48**
- 7 vSphere IaaS Control Plane のバックアップとリストアのトラブルシューティング 52**
  - バックアップからの スーパーバイザー のリストア後の実体のないオブジェクトのクリーンアップ 52

# vSphere IaaS 制御プレーンのバックアップとリストア

『vSphere IaaS 制御プレーンのバックアップとリストア』では、スーパーバイザー 制御プレーンと、Tanzu Kubernetes Grid クラスターおよび vSphere ポッド で実行されているワークロードのバックアップとリストア方法に関する情報を提供します。

## 対象読者

この情報は、vSphere IaaS Control Plane で実行されているワークロードとスーパーバイザー 制御プレーンの状態をバックアップおよびリストアする vSphere 管理者および DevOps エンジニアを対象としています。次の領域に関する知識が必要です。

- vSphere IaaS Control Plane
- vSphere
- Kubernetes
- Velero
- インスタンス ストレージ

# vSphere IaaS Control Plane のバックアップとリストアに関する考慮事項

# 1

vSphere IaaS Control Plane のバックアップおよびリストア プロセスについて説明し、vSphere IaaS Control Plane のバックアップとリストアの戦略を実装するための大局的な考慮事項について理解します。

使用例	Tools	コメント
スーパーバイザー 制御プレーンのバックアップとリストア	ワークロード管理ユーザー インターフェイスからの vCenter Server ファイルベースのバックアップとリストア	<p>vCenter Server のスケジュール設定されたファイルベースのバックアップの一部として、vCenter Server でスーパーバイザーの状態のバックアップを構成します。後で、vSphere Client のワークロード管理インターフェイスを使用して、vCenter Server のスーパーバイザーの状態をリストアできます。</p> <p><b>注:</b> vCenter Server でのスーパーバイザーの状態のリストアと、vCenter Server の状態のリストアは、2つの異なるワークフローです。vCenter Server をリストアしても、スーパーバイザーがホストにリストアされることはありません。</p> <p><a href="#">2章 スーパーバイザー 制御プレーンのバックアップとリストア</a>を参照してください。</p>
vSphere ポッドのバックアップとリストア	Velero Plugin for vSphere	<p>スーパーバイザーにプラグインをインストールして構成します。</p> <p><a href="#">3章 スーパーバイザーでの Velero Plugin for vSphere のインストールと構成</a>を参照してください。</p> <p><a href="#">6章 Velero Plugin for vSphere を使用した vSphere ポッドのバックアップとリストア</a>を参照してください。</p>

使用例	Tools	コメント
Tanzu Kubernetes Grid クラスタ上のステートレスおよびステートフル ワークロードのバックアップと Tanzu Kubernetes Grid でプロビジョニングされているクラスタへのリストア。	Velero Plugin for vSphere	<p>Kubernetes メタデータとパーシステント ボリュームの両方をバックアップおよびリストアします。</p> <p>パーシステント ボリュームには、Velero スナップショット作成を使用できます (Restic は使用できません)。</p> <p><a href="#">3 章 スーパーバイザー での Velero Plugin for vSphere のインストールと構成</a>を参照してください。</p> <p><a href="#">Velero Plugin for vSphere を使用した TKG 2 クラスタ ワークロードのバックアップとリストア</a>を参照してください。</p>
Tanzu Kubernetes Grid クラスタ上のステートレスおよびステートフル ワークロードのバックアップと Tanzu Kubernetes Grid でプロビジョニングされていない適合する Kubernetes クラスタへのリストア。	スタンドアローン Velero と Restic	<p>移植可能にするためにスタンドアローンの Velero を使用します。ステートフル アプリケーションの場合は、Restic を含める必要があります。</p> <p><a href="#">スーパーバイザー上の TKG 2 クラスタへのスタンドアローン Velero と Restic のインストールと構成</a>を参照してください。</p> <p><a href="#">スタンドアローンの Velero と Restic を使用したスーパーバイザー上の TKG 2 クラスタでのワークロードのバックアップとリストア</a>を参照してください。</p>
vCenter Server の構成	vCenter Server	<p>vCenter Server が失われた場合は、vCenter Server を使用して vCenter Server オブジェクトをバックアップおよびリストアします。</p> <p><a href="#">vCenter Server のファイルベースのバックアップとリストア</a>を参照してください。</p>
NSX	NSX Manager	<p>ロード バランサと Ingress サービスは、NSX のバックアップに依存します。</p> <p>NSX-T Data Center が製品内に備えているバックアップとリカバリでは、NSX Manager ノードおよびオブジェクトのバックアップとリストアがサポートされています。詳細については、NSX-T ドキュメントの <a href="#">Backing Up and Restoring NSX Manager</a> を参照してください。</p>

# スーパーバイザー 制御プレーンのバックアップとリストア

# 2

vCenter Server ファイルベースのバックアップの一部として、スーパーバイザー の状態を記録するオプションを vCenter Server に含めることができます。後で、作成したバックアップ ファイルから スーパーバイザー 制御プレーンをリストアできます。

次のトピックを参照してください。

- スーパーバイザー の状態のバックアップ
- スーパーバイザー 制御プレーンのリストア

## スーパーバイザー の状態のバックアップ

環境内の スーパーバイザー の状態をバックアップする方法について説明します。vCenter Server ファイルベースのバックアップの一部として、vCenter Server で使用可能な スーパーバイザー のバックアップを含めることができます。

スーパーバイザー 制御プレーンのバックアップ ファイルは、次のコンポーネントの状態をキャプチャします。

- etcd の状態。
- vCenter Server のアップグレード後に制御プレーン仮想マシンを確実にリストアするためにインフラストラクチャ ボードに使用されるコンテナ イメージ。
- 同じ CA からのリストア後にすべての Kubernetes 証明書を再生成できるようにするための Kubernetes CA 証明書とキー。これにより、新しい Kubernetes CA を信頼するために、リストア後に vSphere ボードと Spherelet を再構成する必要がなくなります。
- すべての vSphere 名前空間 と、デプロイ、ポッド、仮想マシン、TKG リソース、パーシステント ボリュームの要求など、ワークロードに関連付けられているすべての Kubernetes リソースの状態。

vCenter Server ファイルベースのバックアップとリストアの詳細については、「[vCenter Server のファイルベースのバックアップとリストア](#)」を参照してください。

### 前提条件

- バックアップを格納するために必要なディスク容量を確保した FTP、FTPS、HTTP、HTTPS、SFTP、NFS、SMB のいずれかのサーバが実行中である必要があります。

## 手順

- 1 Web ブラウザで、vCenter Server 管理インターフェイス (<https://appliance-IP-address-or-FQDN:5480>) に移動します。
- 2 root としてログインします。
- 3 vCenter Server の管理インターフェイスで、[バックアップ] をクリックします。
- 4 バックアップ スケジュールがすでに作成されている場合は [編集] をクリックします。  
バックアップ スケジュールがない場合は、「[ファイルベースのバックアップのスケジュール設定](#)」を参照して作成方法を確認してください。
- 5 [バックアップ スケジュールの編集] ペインで、[スーパーバイザー制御プレーン] を選択します。

### Edit Backup Schedule ×

Backup location * <span style="font-size: small;">(i)</span>	<input type="text" value="sftp://[redacted]/root/backup"/>	
Backup server credentials *	User name	<input type="text" value="root"/>
	Password	<input type="password"/>
Schedule <span style="font-size: small;">(i)</span>	Weekly <span style="font-size: small;">v</span> Sunday <span style="font-size: small;">v</span>	<input type="text" value="11 : 59 P.M. Etc/UTC"/>
Encrypt backup	Encryption Password	<input type="password"/>
	Confirm Password	<input type="password"/>
Number of backups to retain *	<input checked="" type="radio"/> Retain all backups <input type="radio"/> Retain last <input type="text" value="0"/> backups	
Data	<input checked="" type="checkbox"/> Supervisors Control Plane	909 MB
	<input checked="" type="checkbox"/> Stats, Events, and Tasks	90 MB
	<input checked="" type="checkbox"/> Inventory and configuration	296 MB
	Total size (compressed)	

## 結果

vCenter Server のすべての スーパーバイザー の状態は、vCenter Server バックアップの一部としてバックアップされます。

## スーパーバイザー 制御プレーンのリストア

vCenter Server システム自体のバックアップ ファイルから、vCenter Server のスーパーバイザー 制御プレーンをリストアできます。

**注：** vCenter Server でのスーパーバイザー 制御プレーンのリストアと vCenter Server の状態のリストアは、2つの異なるワークフローです。vCenter Server をリストアしても、スーパーバイザー 制御プレーンはリストアされません。

### 前提条件

- vCenter Server 管理インターフェイスのファイルベースのバックアップからのスーパーバイザー 状態の記録を構成します。

### 手順

- 1 vSphere Client で、[ワークロード管理] に移動します。
- 2 [スーパーバイザー] を選択し、[リストア] を選択します。
- 3 バックアップの詳細を入力します。

オプション	説明
vCenter Server	スーパーバイザー を管理する vCenter Server システムを選択します。
バックアップの選択項目	<ul style="list-style-type: none"> <li>■ バックアップ サーバ フォルダの [参照] を選択して、この vCenter Server システムで構成されたバックアップ ファイル サーバのルート フォルダに保存されているファイルをアップロードします。</li> <li>■ バックアップの場所に [単一] を選択して、特定のバックアップ ファイルをアップロードしてから、そのバックアップ ファイルの URL を入力します。</li> <li>■ [vCenter バックアップ スケジュールに指定されたバックアップの場所とユーザー名を使用します] を選択して、vCenter Server で構成されているバックアップの場所のルート フォルダの場所とユーザー名を入力します。</li> </ul>
場所	バックアップ ルート フォルダの場所を入力します。
ユーザー名	バックアップにアクセスするためのユーザー名を入力します
Password	ユーザー名に対応するパスワードを入力します。

- 4 [次へ] をクリックします。
- 5 リストア元のバックアップ ファイルを選択し、[次へ] をクリックしてバックアップ ファイルのダウンロードを開始します。
- 6 リストアするスーパーバイザー を選択し、[次へ] をクリックします。
- 7 設定内容を確認して、[終了] をクリックします。

### 結果

スーパーバイザー が構成状態に戻り、すべての制御プレーン仮想マシンが削除され、バックアップ ファイルのデータを使用して再デプロイされました。プロセスを監視するには、[構成ステータス] 列で [表示] をクリックします。

# スーパーバイザー での Velero Plugin for vSphere のインストールと構成

# 3

vSphere ポッド および TKG クラスタで実行されているワークロードをバックアップおよびリストアするために Velero Plugin for vSphere をインストールして構成する方法について説明します。

## 概要

Velero Plugin for vSphere は、vSphere IaaS Control Plane ワークロードをバックアップおよびリストアするためのソリューションを提供します。スーパーバイザー に Velero Plugin for vSphere をインストールして構成すると、TKG クラスタ ワークロードと vSphere ポッド をバックアップおよびリストアできるようになります。パーシステント ワークロードの場合は、Velero Plugin for vSphere でパーシステント ボリュームのスナップショットを作成できます。

## 前提条件:

- スーパーバイザー が有効になっている。
- vSphere 名前空間 が作成され、構成されている。
- vSphere 管理者ロールのメンバーであるか、次の vSphere 権限を持っている必要がある。
  - SupervisorServices.Manage
  - Namespaces.Manage
  - Namespaces.Configure

---

**注：** 『vSphere IaaS 制御プレーンでの TKG サービスの使用』の「[専用グループとロールの作成](#)」を参照してください。

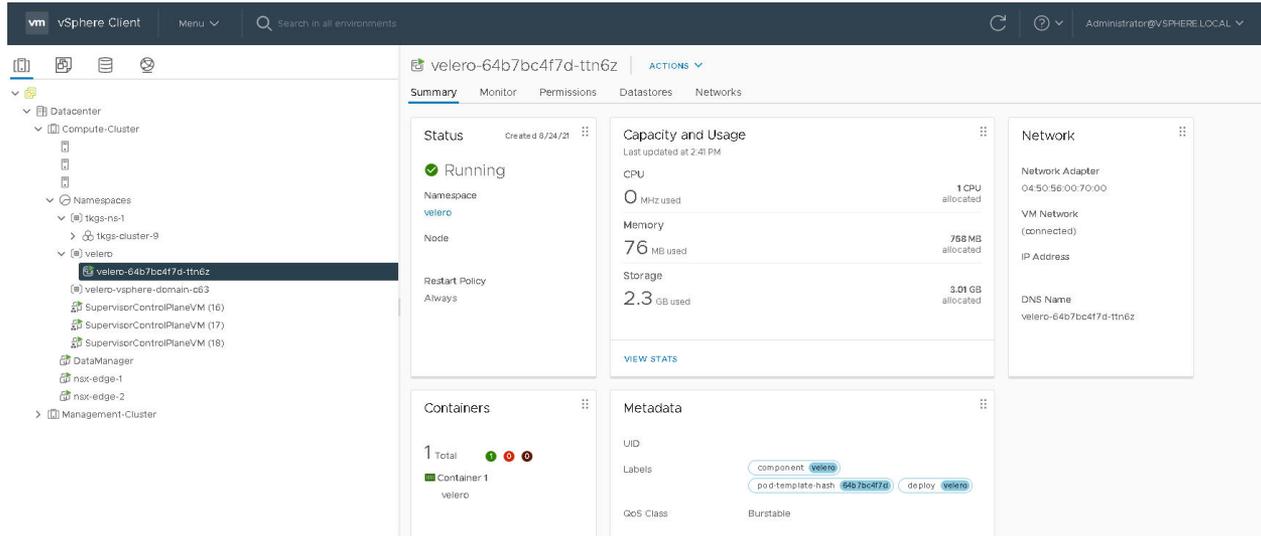
---

- Velero CLI を実行できる Linux 仮想マシンを作成する。または、既存の Linux ジャンプ ホストを使用してスーパーバイザー にアクセスする。
- Velero バージョン番号は `x.y.z` として表されます。使用するバージョンについては、[Velero 互換性マトリクス](#)を参照し、コマンドの実行時に適宜置き換えてください。

このスクリーンショットは、Velero Plugin for vSphere インストールの終了状態を示しています。

- NSX ネットワークは、vSphere ポッド のデプロイをサポートするために使用されている。
- Data Manager 仮想マシンがデプロイされている。
- Velero Operator が有効で、`velero-vsphere-domain-cxx` 名前空間で実行されている。

- velero という名前空間が構成されている。
- Velero Plugin for vSphere が velero 名前空間内で vSphere ポッド として実行されている。



## 手順 0 (オプション) : バックアップおよびリストア トラフィックの専用ネットワークの作成

本番環境でバックアップトラフィックとリストアトラフィックを vSphere IaaS Control Plane 管理ネットワークトラフィックから分離することは、必須ではありませんが、推奨です。この操作には、2つの側面があります。

- ネットワーク ファイル コピー (NFC) をサポートする ESXi ホストにタグを付ける
- NSX を使用してバックアップおよびリストア ネットワークを構成する。

専用ネットワーク ブロック デバイス (NBD) の転送をサポートするように ESXi ホストを構成するには、スーパーバイザー が実行されている vSphere クラスタ内の各 ESXi ホストに VMkernel NIC を追加して、その NIC に vSphereBackupNFC を設定します。タグ vSphereBackupNFC を VMkernel アダプタの NIC タイプに適用すると、バックアップトラフィックとリストアトラフィックは選択した仮想 NIC を経由するようになります。

この構成を実行するには、Virtual Disk Development Kit を使用します。[NBD のドキュメント](#)を参照してください。

**注：** vSphereBackupNFC が VMkernel NIC で有効でない場合は、バックアップとリストアのネットワークが構成されているときでも、バックアップとリストアのトラフィックはこれらのネットワークを経由しません。vSphereBackupNFC が有効でない場合、トラフィックは vSphere 管理ネットワークを経由します。

vSphereBackupNFC タグを有効にしたら、クラスタの既存の vSphere Distributed Switch (VDS) を次のように更新して、NSX を使用してバックアップおよびリストア ネットワークを構成します。

- vSphere Client で [メニュー] - [ネットワーク] の順に選択します。
- クラスタの既存の Distributed Switch を選択します。

- Distributed Switch を右クリックして、[分散ポート グループ] - [新規分散ポート グループ] の順に選択します。
- [BackupRestoreNetwork] という名前の新しい分散ポート グループを作成します。
- [BackupRestoreNetwork] 分散ポート グループに VMkernel アダプタを追加します。
- ワークロード管理が有効になっている vCenter Server クラスタ内のすべての ESXi ホストを [BackupRestoreNetwork] 分散ポート グループに接続します。
- vSphereBackupNFC タグを有効にします。

## 手順 1 : S3 互換オブジェクト ストアの作成

パーシステント ボリュームのバックアップおよびリストアを行うには、S3 互換オブジェクト ストアを指定する必要があります。Velero は、複数のオブジェクト ストア プロバイダをサポートしています。

Velero Plugin for vSphere をインストールするには、S3 互換オブジェクト ストアに関する次の情報を指定する必要があります。

データ項目	値の例
s3Url	http://my-s3-store.example.com
aws_access_key_id	ACCESS-KEY-ID-STRING
aws_secret_access_key	SECRET-ACCESS-KEY-STRING

次の情報を使用して、s3-credentials という名前のシークレット ファイルを作成します。このファイルは、Velero Plugin for vSphere をインストールするときに参照します。

```
[default]
aws_access_key_id = ACCESS-KEY-ID-STRING
aws_secret_access_key = SECRET-ACCESS-KEY-STRING
```

MinIO は、インストールと使用が容易な S3 互換オブジェクト ストアです。vSphere IaaS Control Plane には、ユーザーが有効にできる MinIO スーパーバイザー サービスが付属しています。詳細については、『vSphere IaaS 制御プレーンのサービスとワークロード』のドキュメントを参照してください。

Linux 仮想マシンに MinIO サーバを手動でインストールすることもできます。手順については、TKG クラスタへのスタンドアロン Velero と Restic のインストールと構成を参照してください。

## 手順 2 : Data Manager のインストールと構成

**注意:** Data Manager は機能テストのみが実施されており、大規模な作業を意図したものではなく、パフォーマンスの期待についてなんらかの保証をするものではありません。本番環境でビジネス クリティカルなアプリケーションをバックアップすることを目的としていません。

Velero Plugin for vSphere を使用してバックアップとリストアを容易にするには、1 台以上の Data Manager 仮想マシンをデプロイして、S3 互換オブジェクトストレージとの間でパーシステント ポリリューム バックアップ データを移動します。Data Manager は、バックアップ時には vSphere ポリリュームから耐久性のあるリモート S3 互換ストレージにポリリューム スナップショット データを移動し、リストア時にはリモート S3 互換ストレージから vSphere ポリリュームに移動します。

vSphere IaaS Control Plane 環境で、Data Manager を仮想マシンとしてインストールします。

---

**注：** Velero vSphere Operator を有効にするまで、Data Manager 仮想マシンをパワーオンしないでください。

---

- 1 vSphere Client を使用して、スーパーバイザー が有効になっているデータセンターを右クリックし、[OVF テンプレートのデプロイ] を選択します。
- 2 URL <https://vmwareasas.jfrog.io/artifactory/Velero-YAML/Velero/DataManager/1.2.0/datamgr-ob-20797900-photon-3-release-1.2.ova> から Data Manager OVA ファイルをローカル マシンにダウンロードします。
- 3 [ローカル ファイル] を選択し、Data Manager OVA を vCenter Server にアップロードします。
- 4 たとえば、仮想マシンに **DataManager** という名前を付けます。
- 5 コンピューティング リソースとして、スーパーバイザー が構成されている vSphere クラスタを選択します。
- 6 仮想マシンのデプロイの詳細を確認し、[次へ] をクリックします。
- 7 使用許諾契約書を承諾して、[次へ] をクリックします。
- 8 ストレージを選択して、[次へ] をクリックします。
- 9 Data Manager 仮想マシンのターゲット ネットワークを選択します。
  - 専用のバックアップおよびリストア ネットワークが構成されている場合は、[BackupRestoreNetwork] を選択します。
  - 専用のバックアップおよびリストア ネットワークが構成されていない場合は、[管理ネットワーク] を選択します。
- 10 選択内容を確認し、[終了] をクリックしてプロセスを完了します。
- 11 [最近のタスク] パネルを使用して、デプロイの進行状況を監視します。

---

**注：** 「OVF 記述子が利用できません。」というエラーが表示される場合は、Chrome ブラウザを使用します。

---

- 12 Data Manager 仮想マシンがデプロイされたら、仮想マシンの入力パラメータを構成します。
- 13 仮想マシンを右クリックして、[設定の編集] を選択します。
- 14 [仮想ハードウェア] タブで、[CD/DVD ドライブ] を [ホスト デバイス] から [クライアント デバイス] に変更します。

---

**注：** この操作を行わないと、必要な詳細設定を保存できません。

---

- 15 [設定の編集] - [詳細パラメータ] タブで [詳細] - [構成パラメータの編集] の順に選択します。

16 次の各設定の入力パラメータを構成します。

パラメータ	値
questinfo.cnsdp.vcUser	仮想マシンのデプロイに必要な権限を持っている vCenter Server ユーザーの名前を入力します。 vSphere 管理者権限を持つユーザーを指定しない場合は、 <a href="#">vSphere 権限のドキュメント</a> の説明を参照してください。または、ワークロード管理専用のユーザーを作成します。『vSphere IaaS 制御プレーンでの TKG サービスの使用』の「 <a href="#">専用グループとロールの作成</a> 」を参照してください。
questinfo.cnsdp.vcAddress	vCenter Server の IP アドレスまたは FQDN を入力します。
questinfo.cnsdp.vcPasswd	vCenter Server ユーザー パスワードを入力します。
questinfo.cnsdp.vcPort	デフォルトは <b>443</b> です。この値は変更しないでください。
questinfo.cnsdp.wcpControlPlaneIP	スーパーバイザー のフローティング IP アドレスを入力します。 この値を取得するには、ワークロード管理のスーパーバイザー に移動し、[構成] - [ネットワーク] - [管理ネットワーク] - [フローティング IP アドレス] を選択します
questinfo.cnsdp.updateKubect1	デフォルトは <b>false</b> です。この値は変更しないでください。
questinfo.cnsdp.veleroNamespace	デフォルト値 <code>velero</code> を保持します。プロセスの後半で、スーパーバイザー に <code>velero</code> という名前の vSphere 名前空間を作成します。これら 2 つの名前が一致する必要があります。
questinfo.cnsdp.datamgrImage	構成されていない（未設定の）システムでは、デフォルトで、 <code>vsphereveleroplugin/data-manager-for-plugin:1.1.0</code> にある Docker ハブからコンテナ イメージが取得されます。

17 [OK] をクリックして構成を保存し、もう一度 [OK] をクリックして仮想マシンの設定を保存します。

**注：** [CD/DVD ドライブ] を [ホスト デバイス] から [クライアント デバイス] に変更しなかった場合は、設定を保存できません。この場合は、操作をキャンセルしてドライブを変更し、詳細設定を繰り返します。

18 Velero vSphere Operator を有効にするまで（次のセクションまで）、Data Manager 仮想マシンをパワーオンしないでください。

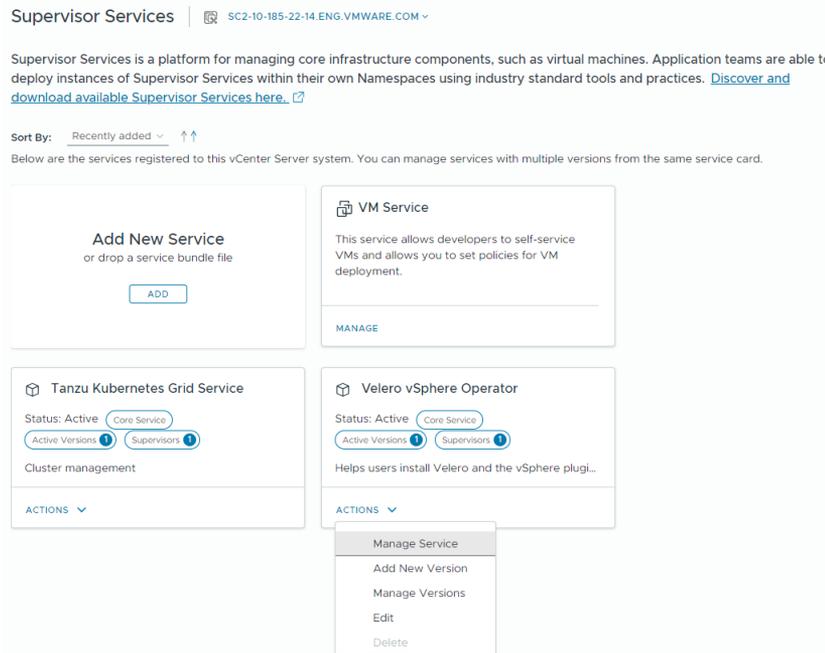
## 手順 3：スーパーバイザー への Velero vSphere Operator サービスのインストール

vSphere IaaS Control Plane は、Velero vSphere Operator をスーパーバイザー サービス として提供します。Velero vSphere Operator サービスは Velero Plugin for vSphere と連携して、パーシステント ボリュームのスナップショット作成など、Kubernetes ワークロードのバックアップとリストアをサポートします。スーパーバイザー サービスの詳細については、『vSphere IaaS 制御プレーンのサービスとワークロード』の「[スーパーバイザー サービスの管理](#)」を参照してください。

Velero vSphere Operator はコア スーパーバイザー サービス です。これは、サービス オペレータが vCenter Server に事前に登録されていることを意味します。スーパーバイザー にサービスとして Velero vSphere Operator をインストールする手順を完了します。

1 vSphere Client ホーム メニューから、[ワークロード管理] を選択します。

- [サービス] タブを選択します。
- 上部のドロップダウンメニューからターゲット vCenter Server を選択します。
- Velero vSphere Operator カードで、[アクション] - [サービスの管理] の順に選択します。



- サービスをインストールするターゲット スーパーバイザー を選択して、[次へ] をクリックします。
- [完了] をクリックして、サービスのインストールを完了します。

スーパーバイザー で Velero vSphere Operator サービスを確認し、Data Manager 仮想マシンを起動します。

- vSphere Client ホーム メニューから、[ワークロード管理] を選択します。
- [サービス] を選択します。
- Velero vSphere Operator がインストールされていて、ステータスが [設定済み] になっていることを確認します。
- [名前空間] タブに、svc-velero-vsphere-domain-xxx という名前の新しい vSphere 名前空間 が表示されていることを確認します。xxx は一意の英数字トークンです。これは、Velero vSphere Operator 用にシステムによって作成された名前空間です。

**注：** この名前空間を構成する必要はありません。編集しないでください。

- [ホストおよびクラスタ] から、Data Manager 仮想マシンを見つけて、仮想マシンをパワーオンします。

## 手順 4 : Velero Plugin for vSphere の vSphere 名前空間 の作成

vSphere Client を使用して、スーパーバイザー で vSphere 名前空間 を手動で作成します。Velero Plugin for vSphere を使用するには、この vSphere 名前空間 が必要です。

- vSphere 名前空間 に **velero** という名前を付けます。

- [velero] 名前空間を選択して、構成します。
- [velero] 名前空間のストレージを指定します。
- 適切な権限を持つユーザーに、[velero] 名前空間に対する編集権限を付与します。

## 手順 5 : Velero Plugin for vSphere 構成マップの作成

velero-vsphere-plugin-config.yaml という名前の Velero Plugin for vSphere の構成マップを作成します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: velero-vsphere-plugin-config
data:
  cluster_flavor: SUPERVISOR
```

構成マップを スーパーバイザー に適用します。

```
kubectl apply -n <velero-namespace> -f velero-vsphere-plugin-config.yaml
```

構成マップをインストールしなかった場合は、Velero Plugin for vSphere をインストールするときに次のエラーが表示されます。

```
Error received while retrieving cluster flavor from config, err: configmaps "velero-vsphere-plugin-config" not found
Falling back to retrieving cluster flavor from vSphere CSI Driver Deployment
```

## 手順 6 : Velero Plugin for vSphere のインストール

これで、Velero Plugin for vSphere をインストールする準備ができました。インストールするには、[velero-vsphere] CLI をダウンロードして実行します。

**注：** この手順には Linux 仮想マシンが必要です。kubectl-vsphere と kubectl CLI を実行する Linux ジャンプ ホストに [velero-vsphere] バイナリをダウンロードします。

- 1 Velero Plugin for vSphere CLI をダウンロードします。

互換性マトリックスを確認し、次の場所からターゲット バージョンをダウンロードします：<https://github.com/vmware-tanzu/velero-plugin-for-vsphere/releases>。

**注：** 以降のコマンドで、X.Y.Z をダウンロードした Velero CLI およびプラグインのバージョンに置き換えます。

- 2 CLI を Linux ジャンプ ホストに安全にコピーします。例：

```
pscp -P 22 C:\temp\velero-vsphere-X.Y.Z-linux-amd64.tar.gz ubuntu@10.117.29.131:/home/ubuntu/tanzu
```

- 3 `velero-vsphere` CLI を抽出して、書き込み可能にします。

```
tar -xf velero-vsphere-X.Y.Z-linux-amd64.tar.gz
chmod +x velero-vsphere
```

- 4 CLI をパスに追加します。

```
export PATH="$(pwd)/velero-vsphere-X.Y.Z-linux-amd64:$PATH"
```

- 5 次の内容の `s3-credentials` ファイルを作成します。

```
aws_access_key_id = ACCESS-KEY-ID-STRING
aws_secret_access_key = SECRET-ACCESS-KEY-STRING
```

- 6 S3 互換オブジェクトストアのリージョン、URL、バケット名を取得します。
- 7 `kubectl` 向けの `vSphere` プラグイン を使用して、スーパーバイザー にログインします。
- 8 コンテキストをスーパーバイザー に切り替えます。

```
kubectl config use-context SUPERVISOR-CLUSTER-IP-ADDRESS
```

- 9 次の `velero-vsphere` CLI コマンドを実行して、作成した `[velero]` 名前空間に `Velero Plugin for vSphere` をインストールします。

AWS の `$BUCKET` と `$REGION` の値をエクスポートします。前述のいずれかの手順に従わなかった場合は、シークレット ファイルの名前や場所、手動で作成した `velero` 名前空間の名前などの値を調整します。

```
export BUCKET=example-velero-sv && export REGION=us-east-1

./velero-vsphere install \
  --namespace velero \
  --version vX.X.X \
  --provider aws \
  --plugins harbor-repo.vmware.com/velero/velero-plugin-for-aws:vX.Y.Z,harbor-
repo.vmware.com/velero/velero-plugin-for-vsphere:vX.Y.Z \
  --bucket $BUCKET \
  --secret-file ~/.aws/credentials \
  --snapshot-location-config region=$REGION \
  --backup-location-config region=$REGION
```

**注：** たとえば、`Velero Plugin for vSphere v1.4.0` が使用されている場合、`Velero CLI` のバージョンは `v1.8.1` です。

- 10 `Velero Plugin for vSphere` が正常にインストールされたことを確認します。

インストールが正常に完了すると、次のメッセージが表示されます。

```
Send the request to the operator about installing Velero in namespace velero
```

次のコマンドを実行して、さらに確認します。「完了」と表示されて、バージョンが表示されます。

```
kubectl -n velero get veleroservice default -o json | jq '.status'
```

予期される結果：

```
{
  "enabled": true,
  "installphase": "Completed",
  "version": "v1.8.1"
}
```

**注：** 上記のコマンドでは、ターミナルに送信される JSON 出力をフォーマットする jq ユーティリティがインストールされていることが前提となります。jq がインストールされていない場合は、インストールするか、コマンドのこの部分 (json 以降のすべて) を削除します。

- 11 必要に応じてトラブルシューティングを行います。

インストールに失敗した場合は、インストールを削除して、再試行してください。インストールを削除するには、次のセクションに記載されている手順を記載されている順序で実行します。

## 補足：Velero Plugin for vSphere のアンインストール

次の手順を実行して、Velero Plugin for vSphere をアンインストールします。

- 1 `velero-vsphere` CLI を実行して、Velero Plugin for vSphere をアンインストールします。

```
./velero-vsphere uninstall -n velero
```

- 2 `velero` という名前の vSphere ポッド が削除されたことを確認します。

```
kubectl get pods -n velero
```

ポッドが「終了しています」と表示されている場合は、ポッドが削除されるまで待機してから続行します。

- 3 vSphere Client を使用して、手動で作成した `velero` という名前の vSphere 名前空間 を削除します。

**注：** 名前空間の削除が完了するまで、次の手順に進まないでください。kubectl を使用して、`velero` 名前空間が削除されていることを確認できます (ただし、kubectl を使用して `velero` 名前空間を削除しないでください)。

## 補足：エアギャップ環境への Velero Plugin for vSphere のインストール

エアギャップ環境に Velero Plugin for vSphere をインストールする場合は、カスタマイズされたイメージとともにインストールする必要があります。カスタマイズされたイメージのうち backup-driver および data-manager-for-plugin の対応するイメージが想定されるレジストリ内にあり、Kubernetes クラスタからアクセスできることを確認する必要があります。エアギャップ環境では、Docker ハブ内のリリース済みイメージにアクセスできないため、プライベート レジストリ内にあるカスタマイズ済みイメージが使用されます。

プラグインをインストールするには、次の手順を実行します。

- 1 velero-plugin-for-vsphere、backup-driver、および data-manager-for-plugin のリリース済みイメージをダウンロードします。
- 2 イメージの名前を変更します。つまり、対応する <Registry endpoint and path> および <Version tag> をイメージにタグ付けして、カスタマイズされたリポジトリにアップロードします。
- 3 カスタマイズした velero-plugin-for-vsphere イメージを使用してプラグインをインストールします。

vanilla クラスタに Velero Plugin for vSphere をインストールすると、backup-driver 環境と data-manager-for-plugin デモンセットという 2 つの追加コンポーネントがバックグラウンドでデプロイされます。スーパーバイザー および Tanzu Kubernetes クラスタでは、backup-driver 環境のみがデプロイされます。

velero-plugin-for-vsphere のコンテナ イメージを指定した場合は、イメージ解析メカニズムを使用して対応する backup-driver および data-manager-for-plugin イメージが解析されます。

コンテナ イメージは、次のパターンの形式になります。

```
<Registry endpoint and path>/<Container name>:<Version tag>
```

velero-plugin-for-vsphere コンテナ イメージを指定した場合は、backup-driver および data-manager-for-plugin の対応するイメージのうち、<Registry endpoint and path> および <Version tag> が一致しているものが解析されます。

たとえば、次の velero-plugin-for-vsphere コンテナ イメージについて考えます。

```
abc.io:8989/x/y/.../z/velero-plugin-for-vsphere:vX.Y.Z
```

次に示す backup-driver と data-manager-for-plugin の対応するイメージが取得されると想定されません。

```
abc.io:8989/x/y/.../z/backup-driver:vX.Y.Z
abc.io:8989/x/y/.../z/data-manager-for-plugin:vX.Y.Z
```

- 4 インストールのトラブルシューティングを行います。

backup-driver と data-manager-for-plugin の対応するイメージの解析中に問題やエラーが発生した場合、インストールは Docker ハブの公式 velero-vsphereplugin リポジトリ内の対応するイメージにフォールバックされます。次の問題が発生すると、フォールバック メカニズムがトリガされます。

- a ユーザー入力内のカスタマイズされた velero-plugin-for-vsphere イメージに、予期しないコンテナ名が使用されています。

たとえば、`x/y/velero-velero-plugin-for-vsphere:vX.Y.Z` が使用されています。

- b Velero のデプロイ名が、`velero` 以外の名前にカスタマイズされています。たとえば、Velero をデプロイする前に、Velero manifests ファイルで Velero のデプロイ名が `velero-server` に更新された場合は、問題がトリガされます。

`velero-plugin-for-vsphere` の既存のイメージ解析メカニズムで認識できるのは、`velero` という固定された名前を持つ Velero 環境のみです。

# TKG サービス クラスタとワークロードのバックアップとリストア

# 4

TKG サービス クラスタとワークロードをバックアップおよびリストアするには、このセクションを参照してください。

次のトピックを参照してください。

- TKG サービス クラスタとワークロードのバックアップとリストアに関する考慮事項
- Velero Plugin for vSphere を使用した TKG クラスタ ワークロードのバックアップとリストア
- スタンドアローンの Velero と Restic を使用した スーパーバイザー での TKG クラスタ ワークロードのバックアップとリストア
- Velero と CSI スナップショットを使用したバックアップとリストア

## TKG サービス クラスタとワークロードのバックアップとリストアに関する考慮事項

このトピックでは、TKG サービス クラスタで実行されているワークロードをバックアップおよびリストアするための考慮事項について説明します。

### TKG サービス クラスタのバックアップとリストア

TKG クラスタをバックアップおよびリストアするには、スーパーバイザー データベースをバックアップします。これにより、vSphere 名前空間 オブジェクトと TKG クラスタ ノード仮想マシンをリストアできます。

vCenter Server 管理インターフェイスを通じて使用可能な vCenter Server バックアップ機能を使用して、スーパーバイザー のバックアップとリストアを有効にします。詳細については、vSphere IaaS Control Plane のバックアップとリストアに関するドキュメントを参照してください。

---

**注：** スーパーバイザー バックアップは、TKG クラスタ ノード仮想マシンをリストアする場合にのみ使用できません。スーパーバイザー バックアップを使用して、TKG クラスタにデプロイされたワークロードをリストアすることはできません。ワークロードは別途バックアップし、クラスタのリストア後にリストアする必要があります。

---

## TKG サービス クラスタで実行されているワークロードのバックアップとリストア

表に、TKG クラスタで実行されているステートレスおよびステートフル ワークロードをバックアップおよびリストアするためのオプションの概要を示します。

**注：** Kubernetes クラスタをバックアップおよびリストアするときにスタンドアローンの Velero を使用すると、移植性が得られます。つまり、TKG サービス でプロビジョニングされていない Kubernetes クラスタにクラスタワークロードをリストアできるようにする場合は、スタンドアローンの Velero を使用することを検討してください。

使用例	ツール	コメント
スーパーバイザー上の TKG クラスタでステートレス ワークロードおよびステートフル ワークロードをバックアップし、スーパーバイザー上の TKG クラスタにリストアします。	Velero Plugin for vSphere Velero Plugin for vSphere を使用した TKG クラスタ ワークロードのバックアップとリストアを参照してください。	Kubernetes メタデータとパーシステント ボリュームの両方をバックアップおよびリストアできます。 Velero スナップショット作成は、ステートフル アプリケーションでパーシステント ボリュームに使用されます。 Velero Plugin for vSphere もスーパーバイザーにインストールされ、構成されている必要があります。
スーパーバイザー上の TKG クラスタでステートレスおよびステートフル ワークロードをバックアップし、適合する Kubernetes クラスタにリストアします。	スタンドアローン Velero と Restic 『スタンドアローンの Velero と Restic を使用したスーパーバイザーでの TKG クラスタワークロードのバックアップとリストア』を参照してください。	Kubernetes メタデータとパーシステント ボリュームの両方をバックアップおよびリストアできます。 Restic は、ステートフル アプリケーションでパーシステント ボリュームに使用されます。 移植性が必要な場合は、このアプローチを使用します。
スーパーバイザー上の TKG クラスタでステートレスおよびステートフル ワークロードをバックアップし、適合する Kubernetes クラスタにリストアします。	CSI スナップショットを使用したスタンドアローン Velero 『Velero と CSI スナップショットを使用したバックアップとリストア』を参照してください。	vSphere 8.0 には、vSphere 8.0 U2 以降および TKr v1.26 以降が必要です。

## Velero Plugin for vSphere を使用した TKG クラスタ ワークロードのバックアップとリストア

このセクションでは、Velero Plugin for vSphere を使用してスーパーバイザー で実行されている TKG クラスタ ワークロードをバックアップおよびリストアする方法について説明します。

### TKG クラスタでの Velero Plugin for vSphere のインストールと構成

Velero Plugin for vSphere を使用して TKG クラスタで実行されているワークロードをバックアップおよびリストアするには、このクラスタに Velero Plugin for vSphere をインストールします。

## 概要

Velero Plugin for vSphere は、TKG サービス クラスタのワークロードをバックアップおよびリストアするためのソリューションを提供します。パーシステント ワークロードの場合は、Velero Plugin for vSphere でパーシステント ボリュームのスナップショットを作成できます。

**注：** バックアップおよびリストアする TKG サービス クラスタ ワークロードに可搬性が必要な場合は、Velero Plugin for vSphere を使用しないでください。Kubernetes クラスタ間で可搬性を確保するには、スタンドアロン Velero と Restic を併用します。

## 前提条件：スーパーバイザー への Velero Plugin for vSphere のインストール

TKGS クラスタに Velero Plugin for vSphere をインストールするには、スーパーバイザー に Velero Plugin for vSphere がインストールされている必要があります。また、スーパーバイザー に NSX ネットワークが構成されている必要があります。3 章 [スーパーバイザー での Velero Plugin for vSphere のインストールと構成](#)を参照してください。

## ストレージ要件

TKG サービス クラスタのバックアップを作成するには、ここに記載されているストレージ バックエンドが必要です。複数のクラスタをバックアップする場合は、異なるクラスタ バックアップに同じストレージ バックエンドを使用しないでください。ストレージ バックエンドを共有すると、バックアップ オブジェクトが同期されます。データのエスケープを回避するには、別のストレージ バックエンドを使用する必要があります。

## 手順 1：Linux Workstation への Velero CLI のインストール

Velero CLI は、Velero とのインターフェイスを提供する標準ツールです。Velero CLI は Velero Plugin for vSphere CLI (`velero-vsphere`) よりも高機能で、Tanzu Kubernetes クラスタ ワークロードのバックアップとリストアに必要です。

Linux ワークステーションに Velero CLI をインストールします。理想的なのは、このワークステーションが、vSphere IaaS Control Plane 環境に関連付けられた CLI (`kubectl`、`kubectl-vsphere`、`velero-vsphere` など) を実行するジャンプ ホストと同じであることです。

Velero バージョン番号は `X.Y.Z` として表されます。使用するバージョンについては、[Velero 互換性マトリックス](#) を参照し、コマンドの実行時に適宜置き換えてください。

次の手順を実行して、Velero CLI をインストールします。

- 1 次のコマンドを実行します。

```
$ wget https://github.com/vmware-tanzu/velero/releases/download/vX.Y.Z/velero-vX.Y.Z-linux-amd64.tar.gz
$ gzip -d velero-vX.Y.Z-linux-amd64.tar.gz && tar -xvf velero-vX.Y.Z-linux-amd64.tar
$ export PATH="$ (pwd) /velero-vX.Y.Z-linux-amd64:$PATH"

$ which velero
/root/velero-vX.Y.Z-linux-amd64/velero
```

## 2 Velero CLI がインストールされていることを確認します。

```
velero version

Client:
  Version: vX.Y.Z
```

### 手順 2 : S3 互換バケットの詳細の取得

便宜上、この手順では、スーパーバイザーに Velero Plugin for vSphere をインストールするときに構成したのと同じ S3 互換オブジェクトストアを使用していると仮定します。本番環境では、別のオブジェクトストアを作成できます。

Velero Plugin for vSphere をインストールするには、S3 互換オブジェクトストアに関する次の情報を指定する必要があります。

データ項目	値の例
s3Url	http://my-s3-store.example.com
aws_access_key_id	ACCESS-KEY-ID-STRING
aws_secret_access_key	SECRET-ACCESS-KEY-STRING

次の情報を使用して、s3-credentials という名前のシークレットファイルを作成します。このファイルは、Velero Plugin for vSphere をインストールするときに参照します。

```
aws_access_key_id = ACCESS-KEY-ID-STRING
aws_secret_access_key = SECRET-ACCESS-KEY-STRING
```

### 手順 3 (オプション A) : ラベルを使用した TKG クラスタへの Velero Plugin for vSphere のインストール (新しい方法)

vSphere 8 Update 3 以降を使用している場合は、ラベルを追加して TKG クラスタに Velero Plugin for vSphere を自動的にインストールできます。

- 1 バックアップストレージの場所にアクセス可能であることを確認します。
- 2 Velero vSphere Operator Core スーパーバイザー サービス が有効になっていることを確認します。

```
kubectl get ns | grep velero
svc-velero-domain-c9           Active   18d
```

- 3 velero という名前の Kubernetes 名前空間がスーパーバイザーに作成されていることを確認します。

```
kubectl get ns | grep velero
svc-velero-domain-c9           Active   18d
velero                          Active   1s
```

- 4 Velero Plugin for vSphere スーパーバイザー サービス がスーパーバイザー で有効になっていることを確認します。

```
velero version
Client:
  Version: v1.11.1
  Git commit: bdbe7eb242b0f64d5b04a7fea86d1edbb3a3587c
Server:
  Version: v1.11.1
```

```
kubectl get veleroservice -A
NAMESPACE  NAME      AGE
velero      default  53m
```

```
velero backup-location get
NAME          PROVIDER  BUCKET/PREFIX  PHASE          LAST VALIDATED          ACCESS
MODE  DEFAULT
default  aws      velero         Available      2023-11-20 14:10:57 -0800 PST
ReadWrite true
```

- 5 velero ラベルをクラスタに追加して、ターゲット TKG クラスタに対して Velero を有効にします。

```
kubectl label cluster CLUSTER-NAME --namespace CLUSTER-NS velero.vsphere.vmware.com/
enabled=true
```

**注：** この作業は、クラスタのプロビジョニング時に vSphere 名前空間 から実行します。

- 6 Velero がインストールされていること、およびクラスタの準備ができていることを確認します。

```
kubectl get ns
NAME          STATUS  AGE
...
velero        Active  2m    <--
velero-vsphere-plugin-backupdriver  Active  2d23h
```

```
kubectl get all -n velero
NAME          READY  STATUS  RESTARTS  AGE
pod/backup-driver-5945d6bcd4-gtw9d  1/1    Running  0          17h
pod/velero-6b9b49449-pq6b4          1/1    Running  0          18h
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
deployment.apps/backup-driver  1/1    1           1          17h
```

```

deployment.apps/velero          1/1      1          1          18h
NAME                            DESIRED  CURRENT    READY      AGE
replicaset.apps/backup-driver-5945d6bcd4  1        1          1          17h
replicaset.apps/velero-6b9b49449        1        1          1          18h

```

```

velero version
Client:
  Version: v1.11.1
  Git commit: bdb7eb242b0f64d5b04a7fea86d1edbb3a3587c
Server:
  Version: v1.11.1

```

### 手順 3 (オプション B) : 手動による TKG クラスタへの Velero Plugin for vSphere のインストール (従来の方法)

Velero CLI を使用して、バックアップおよびリストアするターゲットの TKG クラスタに Velero Plugin for vSphere をインストールします。

Velero CLI コンテキストは、kubectl コンテキストに自動的に従います。Velero CLI コマンドを実行してターゲット クラスタに Velero と Velero Plugin for vSphere をインストールする前に、kubectl コンテキストをターゲット クラスタに設定してください。

- 1 kubectl 向けの vSphere プラグイン を使用して、スーパーバイザー での認証を行います。
- 2 kubectl コンテキストをターゲットの TKG クラスタに設定します。

```
kubectl config use-context TARGET-TANZU-KUBERNETES-CLUSTER
```

- 3 TKG クラスタで、Velero プラグイン `velero-vsphere-plugin-config.yaml` の構成マップを作成します。

```

apiVersion: v1
kind: ConfigMap
metadata:
  name: velero-vsphere-plugin-config
data:
  cluster_flavor: GUEST

```

TKG クラスタに構成マップを適用します。

```
kubectl apply -n <velero-namespace> -f velero-vsphere-plugin-config.yaml
```

構成マップをインストールしなかった場合は、Velero Plugin for vSphere をインストールするときに次のエラーが表示されます。

```

Error received while retrieving cluster flavor from config, err: configmaps "velero-vsphere-plugin-config" not found
Falling back to retrieving cluster flavor from vSphere CSI Driver Deployment

```

- 4 次の Velero CLI コマンドを実行して、ターゲット クラスタに Velero をインストールします。

**BUCKET-NAME**、**REGION** (2つのインスタンス)、および **s3Url** フィールドのプレースホルダ値を適切な値に置き換えます。前述のいずれかの手順に従わなかった場合は、シークレット ファイルの名前や場所、手動で作成した `velero` 名前空間の名前などの値を調整します。

```
./velero install --provider aws \
--bucket BUCKET-NAME \
--secret-file ./s3-credentials \
--features=EnableVSPHEREItemActionPlugin \
--plugins velero/velero-plugin-for-aws:vX.Y.Z \
--snapshot-location-config region=REGION \
--backup-location-config region=REGION,s3ForcePathStyle="true",s3Url=http://my-s3-
store.example.com
```

- 5 ターゲット クラスタに Velero Plugin for vSphere をインストールします。インストールされている Velero は Kubernetes API サーバと通信して、プラグインをインストールします。

```
velero plugin add vsphereveleroplugin/velero-plugin-for-vsphere:vX.Y.Z
```

## 補足 : TKG クラスタからの Velero Plugin for vSphere のアンインストール

次の手順を実行して、Velero Plugin for vSphere をアンインストールします。

- 1 `kubectl` コンテキストをターゲット Tanzu Kubernetes クラスタに設定します。

```
kubectl config use-context TARGET-TANZU-KUBERNETES-CLUSTER
```

- 2 プラグインをアンインストールするには、次のコマンドを実行して、Velero のデプロイから `velero-plugin-for-vsphere` の `InitContainer` を削除します。

```
velero plugin remove vsphereveleroplugin/velero-plugin-for-vsphere:vX.Y.Z
```

- 3 プロセスを完了するには、バックアップ ドライバのデプロイと関連する CRD を削除します。

```
kubectl -n velero delete deployment.apps/backup-driver
```

```
kubectl delete crds \
backuprepositories.backupdriver.cnsdp.vmware.com \
backuprepositoryclaims.backupdriver.cnsdp.vmware.com \
clonefromsnapshots.backupdriver.cnsdp.vmware.com \
deletesnapshots.backupdriver.cnsdp.vmware.com \
snapshots.backupdriver.cnsdp.vmware.com
```

```
kubectl delete crds uploads.datamover.cnsdp.vmware.com downloads.datamover.cnsdp.vmware.com
```

## Velero Plugin for vSphere を使用した TKG クラスタ ワークロードのバックアップとリストア

Velero Plugin for vSphere を使用して、スーパーバイザー上の TKG クラスタで実行されているワークロードをバックアップおよびリストアできます。

## 前提条件

Velero Plugin for vSphere を使用して TKG クラスタ ワークロードをバックアップおよびリストアするには、最初に Velero Plugin for vSphere をターゲット クラスタにインストールする必要があります。TKG クラスタでの Velero Plugin for vSphere のインストールと構成を参照してください。

## ワークロードのバックアップ

Velero バックアップを作成するコマンドの例を次に示します。

```
velero backup create <backup name> --include-namespaces=my-namespace
```

Velero バックアップは、すべてのローカル スナップショットが作成され、ボリューム スナップショットを除く Kubernetes メタデータがオブジェクト ストアにアップロードされた後に、Completed としてマークされます。この時点でも、非同期のデータ移動タスク、つまりボリューム スナップショットのアップロードはバックグラウンドで実行されており、完了に時間がかかることがあります。ボリューム スナップショットのステータスは、[スナップショット カスタム リソース \(CR\)](#) を監視することにより確認できます。

## スナップショット

スナップショットは、パーシステント ボリュームをバックアップするために使用されます。スナップショット CR は、スナップショットが作成されたパーシステント ボリューム要求 (PVC) と同じ名前空間に、ボリューム スナップショットごとに作成されます。

PVC 名前空間のすべてのスナップショットを取得するには、次のコマンドを実行します。

```
kubectl get -n <pvc namespace> snapshot
```

スナップショットのカスタム リソース定義 (CRD) には、`.status.phase` フィールドのフェーズが多数あります。次に例を示します。

スナップショット フェーズ	説明
新規	まだ処理されていません
SnapshotCreated	ローカル スナップショットが作成されました
SnapshotFailed	ローカル スナップショットの作成に失敗しました
アップロードしています	スナップショットをアップロードしています
Uploaded	スナップショットがアップロードされました
UploadFailed	スナップショットのアップロードに失敗しました
Canceling	スナップショットのアップロードをキャンセルしています
キャンセルされました	スナップショットのアップロードがキャンセルされました
CleanupAfterUploadFailed	スナップショットのアップロード後のローカル スナップショットのクリーンアップに失敗しました

## ワークロードのリストア

Velero リストアのコマンドの例を次に示します。

```
velero restore create --from-backup <velero-backup-name>
```

Velero リストアが Completed としてマークされるのは、ボリューム スナップショットと他の Kubernetes メタデータが現在のクラスタに正常にリストアされたときです。この時点で、このリストアに関連する vSphere プラグインのすべてのタスクも完了します。Velero バックアップの場合とは異なり、バックグラウンドの非同期のデータ移動タスクはありません。

## CloneFromSnapshot

各ボリューム スナップショットからリストアするために、最初にスナップショットが作成された PVC と同じ名前空間に CloneFromSnapshot カスタム リソース (CR) が作成されます。PVC 名前空間のすべての CloneFromSnapshot を取得するには、次のコマンドを実行します。

```
kubectl -n <pvc namespace> get clonefromsnapshot
```

CloneFromSnapshot CRD には、.status.phase フィールドの主要なフェーズがいくつかあります。

スナップショット フェーズ	説明
新規	スナップショットからのクローン作成が完了していません
InProgress	vSphere ボリュームのスナップショットがリモート リポジトリからダウンロードされています
完了	スナップショットからのクローン作成が完了しました
Failed	スナップショットからのクローン作成に失敗しました

## スタンドアローンの Velero と Restic を使用した スーパーバイザーでの TKG クラスタ ワークロードのバックアップとリストア

このセクションでは、スタンドアローンの Velero と Restic を使用して スーパーバイザー で実行されている TKG クラスタ ワークロードをバックアップおよびリストアする方法について説明します。

### TKG クラスタへのスタンドアローン Velero と Restic のインストールと構成

スーパーバイザー上の TKG クラスタで実行中のワークロードをバックアップおよびリストアするには、データストアを作成し、Kubernetes クラスタに Velero と Restic をインストールします。

#### 概要

TKG クラスタは仮想マシン ノードで実行されます。TKG クラスタ ワークロードをバックアップおよびリストアするには、そのクラスタに Velero と Restic をインストールします。

## 前提条件

Tanzu Kubernetes クラスタで実行されているワークロードをバックアップおよびリストアするために Velero と Restic をインストールするための次の前提条件を環境が満たしていることを確認します。

- いくつかのワークロード バックアップを保存するために十分なストレージを持つ Linux 仮想マシン。この仮想マシンに MinIO をインストールします。
- kubectl 向けの vSphere プラグイン と kubectl を含む vSphere 向け Kubernetes CLI Tools がインストールされる Linux 仮想マシン。このクライアント仮想マシンに Velero CLI をインストールします。このような仮想マシンがない場合は Velero CLI をローカルにインストールできますが、条件に合わせてインストール手順を調整する必要があります。
- Kubernetes 環境がインターネットにアクセスでき、クライアント仮想マシンからアクセスできること。

## MinIO オブジェクト ストアのインストールと構成

Velero は、Kubernetes ワークロードのバックアップ先として S3 互換のオブジェクト ストアを必要とします。Velero は、そのような **オブジェクト ストア プロバイダ** をいくつかサポートしています。簡単にするために、これらの手順では、オブジェクト ストア仮想マシンでローカルに実行される S3 互換のストレージ サービスである **MinIO** を使用します。

- 1 MinIO をインストールします。

```
wget https://dl.min.io/server/minio/release/linux-amd64/minio
```

- 2 MinIO に実行権限を付与します。

```
chmod +x minio
```

- 3 ファイル システムに MinIO 用のディレクトリを作成します。

```
mkdir /DATA-MINIO
```

- 4 MinIO サーバを起動します。

```
./minio server /DATA-MINIO
```

- 5 MinIO サーバが起動すると、エンドポイント URL、AccessKey、SecretKey などの重要なデータストア インスタンスの詳細が得られます。表に示されているエンドポイント URL、AccessKey、SecretKey を記録します。

データストア メタデータ	値
エンドポイント URL	
AccessKey	
SecretKey	

- 6 ブラウザで MinIO サーバのエンドポイント URL を指定して、MinIO データストアを参照します。
- 7 MinIO サーバにログインし、AccessKey と SecretKey を指定します。

- 8 MinIO をサービスとして有効にするには、`minio.service` スクリプトをダウンロードして、MinIO に自動起動を構成します。

```
curl -O https://raw.githubusercontent.com/minio/minio-service/master/linux-systemd/minio.service
```

- 9 `minio.service` スクリプトを編集し、`ExecStart` に次の値を追加します。

```
ExecStart=/usr/local/bin/minio server /DATA-MINIO path
```

- 10 変更したスクリプトを保存します。
- 11 次のコマンドを実行して MinIO サービスを構成します。

```
cp minio.service /etc/systemd/system
cp minio /usr/local/bin/
systemctl daemon-reload
systemctl start minio
systemctl status minio
systemctl enable minio
```

- 12 MinIO ブラウザを起動し、オブジェクトストアにログインして、バックアップとリストア用の MinIO バケットを作成します。
- 13 バケット作成のアイコンをクリックします。
- 14 たとえば `my-cluster-backups` のように、バケット名を入力します。
- 15 バケットが作成されたことを確認します。
- 16 デフォルトでは、新しい MinIO バケットは読み取り専用です。Velero スタンドアローンのバックアップとリストアの場合、MinIO バケットは読み取り/書き込みである必要があります。バケットを読み取り/書き込みに設定するには、バケットを選択し、省略記号（点が並んだ記号）のリンクをクリックします。
- 17 [ポリシーの編集] を選択します。
- 18 ポリシーを [読み取りおよび書き込み] に変更します。
- 19 [追加] をクリックします。
- 20 ダイアログ ボックスを閉じるために、X をクリックします。

## Velero CLI のインストール

仮想マシン クライアントまたはローカル マシンに Velero CLI をインストールします。

このドキュメントで使用されるバージョンは、*Tanzu Kubernetes Grid 2.2.0 用 Velero 1.9.7*です。

- 1 [VMware Customer Connect ポータル](#)の [Tanzu Kubernetes Grid 製品ダウンロード ページ](#)から Velero をダウンロードします。

---

**注：** VMware からサポートを受けるには、VMware によって署名された Velero バイナリを使用する必要があります。

---

- 2 コマンドラインを開き、Velero CLI をダウンロードしたディレクトリに移動します。
- 3 ダウンロード ファイルを解凍します。例：

```
gunzip velero-linux-vX.X.X_vmware.1.gz
```

- 4 Velero バイナリを確認します。

```
ls -l
```

- 5 Velero CLI に実行権限を付与します。

```
chmod +x velero-linux-vX.X.X_vmware.1
```

- 6 Velero CLI をシステム パスに移動して、グローバルに使用可能にします。

```
cp velero-linux-vX.X.X_vmware.1 /usr/local/bin/velero
```

- 7 インストールを確認します。

```
velero version
```

## Tanzu Kubernetes クラスタへの Velero と Restic のインストール

Velero CLI のコンテキストは、kubectl のコンテキストに自動的に従います。Velero CLI コマンドを実行して Velero と Restic をターゲット クラスタにインストールする前に、kubectl のコンテキストを設定します。

- 1 MinIO バケットの名前を取得します。たとえば、my-cluster-backups です。
- 2 MinIO バケットの AccessKey と SecretKey を取得します。
- 3 Velero CLI の動作対象となるクラスタが認識されるように、コンテキストをターゲットの Kubernetes クラスタに設定します。

```
kubectl config use-context tkgs-cluster-name
```

- 4 credentials-minio という名前のシークレット ファイルを作成します。収集した MinIO サーバ アクセス認証情報を使用して、ファイルを更新します。例：

```
aws_access_key_id = 0XXN08JCCGV41QZBV0RQ
aws_secret_access_key = c1Z1bf8Ljkvkmq7fHucrKckxV39BRbcycGeXQDfx
```

**注：** 「バックアップ ストアの取得中にエラーが発生しました」というエラー メッセージと「NoCredentialProviders: チェーン内に有効なプロバイダがありません」という説明が表示された場合は、認証情報ファイルの先頭に [default] 行を追加します。例：

```
[default]
aws_access_key_id = 0XXN08JCCGV41QZBV0RQ
aws_secret_access_key = c1Z1bf8Ljkvkmq7fHucrKckxV39BRbcycGeXQDfx
```

- 5 ファイルを保存し、ファイルが正しい場所に置かれたことを確認します。

```
ls
```

- 6 次のコマンドを実行して、ターゲット Kubernetes クラスタに Velero と Restic をインストールします。両方の URL を MinIO インスタンスの URL に置き換えます。

```
velero install \
--provider aws \
--plugins velero/velero-plugin-for-aws:v1.0.0 \
--bucket tkgs-velero \
--secret-file ./credentials-minio \
--use-volume-snapshots=false \
--use-restic \
--backup-location-config \
region=minio,s3ForcePathStyle="true",s3Url=http://10.199.17.63:9000,publicUrl=http://
10.199.17.63:9000
```

- 7 Velero と Restic のインストールを確認します。

```
kubectl logs deployment/velero -n velero
```

- 8 velero 名前空間を確認します。

```
kubectl get ns
```

- 9 velero ポッドと restic ポッドを確認します。

```
kubectl get all -n velero
```

## Restic DaemonSet のトラブルシューティング（必要な場合）

Kubernetes クラスタで 3 ポッドの Restic DaemonSet を実行するには、Restic DaemonSet の仕様を更新して hostPath を変更することが必要な場合があります。この問題の詳細については、Velero のドキュメントで [Restic Integration](#) を参照してください。

- 1 3 ポッドの Restic DaemonSet を確認します。

```
kubectl get pod -n velero
```

ポッドのステータスが CrashLoopBackOff の場合は、次のように編集します。

- 2 edit コマンドを実行します。

```
kubectl edit daemonset restic -n velero
```

- 3 hostPath を /var/lib/kubelet/pods から /var/vcap/data/kubelet/pods に変更します。

```
- hostPath:
  path: /var/vcap/data/kubelet/pods
```

- 4 ファイルを保存します。
- 5 3 ポッドの Restic DaemonSet を確認します。

```
kubectl get pod -n velero
```

NAME	READY	STATUS	RESTARTS	AGE
restic-5jln8	1/1	Running	0	73s
restic-bpvtq	1/1	Running	0	73s
restic-vg8j7	1/1	Running	0	73s
velero-72c84322d9-1e7bd	1/1	Running	0	10m

## Velero のメモリ制限の調整 (必要な場合)

Velero のバックアップが数時間に渡って `status=InProgress` を返す場合は、メモリ設定の `limits` と `requests` の値を増やします。

- 1 次のコマンドを実行します。

```
kubectl edit deployment/velero -n velero
```

- 2 メモリ設定の `limits` と `requests` をデフォルトの 256Mi および 128Mi から 512Mi および 256Mi に変更します。

```
ports:
- containerPort: 8085
  name: metrics
  protocol: TCP
resources:
  limits:
    cpu: "1"
    memory: 512Mi
  requests:
    cpu: 500m
    memory: 256Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
```

## スタンドアロンの Velero と Restic を使用したクラスタ ワークロードのバックアップとリストア

スタンドアロンの Velero と Restic を使用して、TKG クラスタで実行されているワークロードをバックアップおよびリストアできます。これは Velero Plugin for vSphere を使用する代替の方法です。スタンドアロンの Velero は、主に移植性が必要な場合に利用します。ステートフル ワークロードには、Restic が必要です。

## 前提条件

スタンドアローンの Velero と Restic を使用して TKG クラスタでワークロードをバックアップおよびリストアするには、ターゲット クラスタにスタンドアローン バージョンの Velero と Restic をインストールする必要があります。リストアを別のターゲット クラスタで実行する場合は、ターゲット クラスタにも Velero と Restic をインストールする必要があります。TKG クラスタへのスタンドアローン Velero と Restic のインストールと構成を参照してください。

## TKG クラスタで実行されているステートレス アプリケーションのバックアップ

TKG クラスタで実行されているステートレス アプリケーションをバックアップするには、Velero を使用する必要があります。

この例では、サンプル ステートレス アプリケーションの名前空間にすべてのアプリケーション コンポーネントがある場合に、`--include namespaces` タグを使用して、そのサンプル ステートレス アプリケーションをバックアップおよびリストアする方法を示します。

```
velero backup create example-backup --include-namespaces example-backup
```

次の内容が表示されます。

```
Backup request "example-backup" submitted successfully.
Run `velero backup describe example-backup` or `velero backup logs example-backup` for more
details.
```

作成されたバックアップを確認します。

```
velero backup get
```

```
velero backup describe example-backup
```

MinIO サーバなどの S3 互換オブジェクトストアの Velero バケットを確認します。

Velero は Kubernetes カスタム リソース定義 (CRD) にメタデータを書き込みます。

```
kubectl get crd
```

Velero CRD では、次のような特定のコマンドを実行できます。

```
kubectl get backups.velero.io -n velero
```

```
kubectl describe backups.velero.io guestbook-backup -n velero
```

## TKG クラスタで実行されているステートレス アプリケーションのリストア

TKG クラスタで実行されているステートレス アプリケーションをリストアするには、Velero を使用する必要があります。

サンプル アプリケーションのリストアをテストするために、サンプル アプリケーションを削除します。

名前空間を削除します。

```
kubectl delete ns guestbook
namespace "guestbook" deleted
```

アプリケーションをリストアします。

```
velero restore create --from-backup example-backup
```

次の内容が表示されます。

```
Restore request "example-backup-20200721145620" submitted successfully.
Run `velero restore describe example-backup-20200721145620` or `velero restore logs example-
backup-20200721145620` for more details.
```

アプリケーションがリストアされたことを確認します。

```
velero restore describe example-backup-20200721145620
```

検証のために、次のコマンドを実行します。

```
velero restore get
```

```
kubectl get ns
```

```
kubectl get pod -n example
```

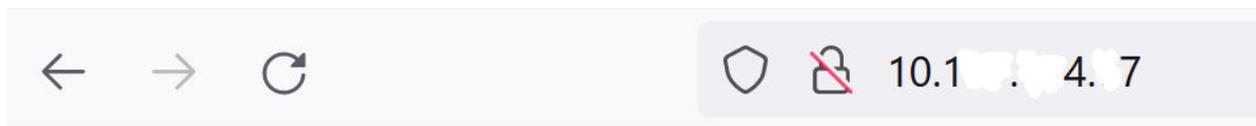
```
kubectl get svc -n example
```

## TKG クラスタで実行されているステートフル アプリケーションのバックアップ

TKG クラスタで実行されているステートフル アプリケーションをバックアップする場合は、保存されているアプリケーション メタデータとアプリケーション データの両方をパーシステント ボリュームにバックアップする必要があります。これを行うには、Velero と Restic の両方が必要です。

この例では、Guestbook アプリケーションを使用します。Guestbook アプリケーションは、TKG クラスタにデプロイされているものと想定します。[#unique\\_17](#) を参照してください。

ステートフルなバックアップとリストアを説明するために、フロントエンド Web ページを使用してメッセージを Guestbook アプリケーションに送信し、そのメッセージが保存されるようにしてみましょう。例：



# Guestbook

Messages

Submit

message 1

message 2

message 3

この例では、`--include namespace` タグやポッドの注釈を使用して、Guestbook アプリケーションをバックアップおよびリストアする方法を示します。

**注：** この例では、注釈を使用します。ただし、Velero バージョン 1.5 以降では、注釈は不要になりました。注釈を使用しないようにするには、バックアップの作成時に `--default-volumes-to-restic` オプションを使用してください。これにより、すべての PV が Restic を使用して自動的にバックアップされます。詳細については <https://velero.io/docs/v1.5/restic/> を参照してください。

バックアップ手順を開始するために、ポッドの名前を取得します。

```
kubectl get pod -n guestbook
```

例：

```
kubectl get pod -n guestbook
```

NAME	READY	STATUS	RESTARTS	AGE
guestbook-frontend-deployment-85595f5bf9-h8cff	1/1	Running	0	55m
guestbook-frontend-deployment-85595f5bf9-lw6tg	1/1	Running	0	55m
guestbook-frontend-deployment-85595f5bf9-wpqc8	1/1	Running	0	55m
redis-leader-deployment-64fb8775bf-kbs6s	1/1	Running	0	55m
redis-follower-deployment-84cd76b975-jrn8v	1/1	Running	0	55m
redis-follower-deployment-69df9b5688-zml4f	1/1	Running	0	55m

パーシステント ボリュームが Redis ポッドに接続されます。これらのステートフル ポッドは、Restic を使用してバックアップします。このため、volumeMount の名前を使用して、ステートフル ポッドに注釈を追加する必要があります。

ステートフル ポッドに注釈を付けるには、volumeMount を知る必要があります。mountName を取得するには、次のコマンドを実行します。

```
kubectl describe pod redis-leader-deployment-64fb8775bf-kbs6s -n guestbook
```

結果には、redis-leader-data からの Containers.leader.Mounts: /data が表示されます。この最後のトークンが、リーダー ポッドの注釈に使用する volumeMount 名です。フォロワーの場合は、redis-follower-data になります。volumeMount 名は、ソース YAML から取得することもできます。

各 Redis ポッドに注釈を付けます。次に例を示します。

```
kubectl -n guestbook annotate pod redis-leader-64fb8775bf-kbs6s backup.velero.io/backup-volumes=redis-leader-data
```

次の内容が表示されます。

```
pod/redis-leader-64fb8775bf-kbs6s annotated
```

注釈を確認します。

```
kubectl -n guestbook describe pod redis-leader-64fb8775bf-kbs6s | grep Annotations
Annotations:  backup.velero.io/backup-volumes: redis-leader-data
```

```
kubectl -n guestbook describe pod redis-follower-779b6d8f79-5dphr | grep Annotations
Annotations:  backup.velero.io/backup-volumes: redis-follower-data
```

Velero バックアップを実行します。

```
velero backup create guestbook-backup --include-namespaces guestbook
```

次の内容が表示されます。

```
Backup request "guestbook-backup" submitted successfully.
Run `velero backup describe guestbook-pv-backup` or `velero backup logs guestbook-pv-backup`
for more details.
```

作成されたバックアップを確認します。

```
velero backup get
```

NAME	STATUS	ERRORS	WARNINGS	CREATED
EXPIRES	STORAGE LOCATION	SELECTOR		
guestbook-backup	Completed	0	0	2020-07-23 16:13:46 -0700 PDT
29d	default	<none>		

バックアップの詳細を確認します。

```
velero backup describe guestbook-backup --details
```

Velero では、他にも次のようなコマンドを実行できます。

```
kubectl get backups.velero.io -n velero
```

```
NAME                AGE
guestbook-backup    4m58s
```

次のコマンドも実行できます。

```
kubectl describe backups.velero.io guestbook-backup -n velero
```

## TKG 2.0 クラスタで実行されているステートフル アプリケーションのリストア

TKG クラスタで実行されるステートフル アプリケーションをリストアする場合は、保存されているアプリケーション メタデータとアプリケーション データの両方をパーシステント ボリュームにリストアする必要があります。これを行うには、Velero と Restic の両方が必要です。

この例では、前のセクションの説明に従ってステートフル Guestbook アプリケーションをバックアップしたものと想定しています。

ステートフル アプリケーションのリストアをテストするために、その名前空間を削除します。

```
kubectl delete ns guestbook
namespace "guestbook" deleted
```

アプリケーションが削除されたことを確認します。

```
kubectl get ns
kubectl get pvc,pv --all-namespaces
```

バックアップからアプリケーションをリストアするには、次のコマンド構文を使用します。

```
velero restore create --from-backup <velero-backup-name>
```

例：

```
velero restore create --from-backup guestbook-backup
```

次のようなメッセージが表示されます。

```
Restore request "guestbook-backup-20200723161841" submitted successfully.
Run `velero restore describe guestbook-backup-20200723161841` or `velero restore logs
guestbook-backup-20200723161841` for more details.
```

ステートフル Guestbook アプリケーションがリストアされたことを確認します。

```

velero restore describe guestbook-backup-20200723161841

Name:          guestbook-backup-20200723161841
Namespace:     velero
Labels:        <none>
Annotations:   <none>

Phase: Completed

Backup: guestbook-backup

Namespaces:
  Included: all namespaces found in the backup
  Excluded: <none>

Resources:
  Included:      *
  Excluded:      nodes, events, events.events.k8s.io, backups.velero.io,
restores.velero.io, resticrepositories.velero.io
  Cluster-scoped: auto

Namespace mappings: <none>

Label selector: <none>

Restore PVs: auto

Restic Restores (specify --details for more information):
  Completed: 3

```

さらに次のコマンドを実行して、リストアを確認します。

```

velero restore get

```

NAME	BACKUP	STATUS	ERRORS	WARNINGS
CREATED	SELECTOR			
guestbook-backup-20200723161841	guestbook-backup	Completed	0	0
2021-08-11 16:18:41 -0700 PDT	<none>			

名前空間がリストアされたことを確認します。

```

kubectl get ns

```

NAME	STATUS	AGE
default	Active	16d
guestbook	Active	76s
...		
velero	Active	2d2h

アプリケーションがリストアされたことを確認します。

```
vkubectl get all -n guestbook
```

NAME	READY	STATUS	RESTARTS	AGE
pod/frontend-6cb7f8bd65-h2pnb	1/1	Running	0	6m27s
pod/frontend-6cb7f8bd65-kwlpr	1/1	Running	0	6m27s
pod/frontend-6cb7f8bd65-snw14	1/1	Running	0	6m27s
pod/redis-leader-64fb8775bf-kbs6s	1/1	Running	0	6m28s
pod/redis-follower-779b6d8f79-5dphr	1/1	Running	0	6m28s
pod/redis-follower-899c7e2z65-8apnk	1/1	Running	0	6m28s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/guestbook-frontend 80:31513/TCP 65s	LoadBalancer	10.10.89.59	10.19.15.99
service/redis-follower 6379/TCP 65s	ClusterIP	10.111.163.189	<none>
service/redis-leader 6379/TCP 65s	ClusterIP	10.111.70.189	<none>

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/guestbook-frontend-deployment	3/3	3	3	65s
deployment.apps/redis-follower-deployment	1/2	2	1	65s
deployment.apps/redis-leader-deployment	1/1	1	1	65s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/guestbook-frontend-deployment-56fc5b6b47	3	3	3	65s
replicaset.apps/redis-follower-deployment-6fc9cf5759	2	2	1	65s
replicaset.apps/redis-leader-deployment-7d89bbdbcf	1	1	1	65s

パーシステント ボリュームがリストアされたことを確認します。

```
kubect1 get pvc,pv -n guestbook
```

NAME	STATUS
VOLUME	CAPACITY ACCESS MODES STORAGECLASS AGE
persistentvolumeclaim/redis-leader-claim a198-5379a2552509 2Gi RWO	Bound thin-disk 2m40s
persistentvolumeclaim/redis-follower-claim b418-2cc680c0560b 2Gi RWO	Bound thin-disk 2m40s

NAME	CAPACITY	ACCESS MODES	RECLAIM
POLICY STATUS CLAIM	STORAGECLASS	REASON	AGE
persistentvolume/pvc-55591938-921f-452a-b418-2cc680c0560b Delete Bound guestbook/redis-follower-claim	2Gi thin-disk	RWO	2m40s
persistentvolume/pvc-a2f6e6d4-42db-4fb8-a198-5379a2552509 Delete Bound guestbook/redis-leader-claim	2Gi thin-disk	RWO	2m40s

最後に、guestbook-frontend サービスの外部 IP アドレスを使用して Guestbook フロントエンドにアクセスし、チュートリアルの最初に送信したメッセージがリストアされたことを確認します。例：

←
→
↻

🛡️
🔒
10.1 . . 4.7

# Guestbook

Messages

Submit

message 1

message 2

message 3

## Velero と CSI スナップショットを使用したバックアップとリストア

Velero と CSI スナップショットを使用して、スーパーバイザー にプロビジョニングされた TKG クラスタで実行されているワークロードに関して CSI で作成されたパーシステント ボリュームをバックアップおよびリストアできます。

### 要件

次の要件を満たす必要があります。

- vSphere 8.0 U2 以降
- vSphere 8.x 以降の場合、Tanzu Kubernetes リリース v1.26.5
- ボリューム スナップショットをサポートする CSI ドライバを使用して作成されたパーシステント ボリューム

**注目:** Velero と CSI スナップショットを使用できるのは、ボリューム スナップショットをサポートする CSI ドライバを使用して作成されたパーシステント ボリュームの場合のみです。詳細については、『vSphere IaaS 制御プレーンでの TKG サービスの使用』の「TKG クラスタでのスナップショットの作成」を参照してください。

## 手順

Velero とコンテナ ストレージ インターフェイス (CSI) スナップショットを使用して、TKGS クラスタで実行されているワークロードをバックアップおよびリストアできます。Velero ノードエージェントは、CSI スナップショットのデータ移動を使用してバックアップとリストアの具体的なタスクを実行するモジュールをホストする DaemonSet です。詳細については、「[Container Storage Interface Snapshot Support in Velero](#)」を参照してください。

- 1 MinIO や AWS S3 バケットなどの S3 互換ストレージの場所を作成します。

次の例では、AWS S3 バケットを使用しています。

MinIO を使用するには、「[MinIO オブジェクト ストアのインストールと構成](#)」を参照してください。

- 2 kubectl を実行しているクラスタ クライアントに Velero CLI をインストールします。

<https://github.com/vmware-tanzu/velero/releases> からダウンロードします。

次のいずれかのリンクからインストール手順を参照してください。

- [手順 1: Linux Workstation への Velero CLI のインストール](#)
- [Velero CLI のインストール](#)
- <https://velero.io/docs/v1.12/basic-install/#install-the-cli>

- 3 Velero バックアップを実行する TKG サービス クラスタに接続します。

「[Kubectl を使用した vCenter Single Sign-On ユーザーとしての TKG サービス クラスタへの接続](#)」を参照してください。

- 4 Velero を Velero CLI プラグインとともにクラスタにインストールします。

Velero v1.14 リリース以降、Velero CSI プラグインは Velero とマージされています。したがって、Velero v1.14 以降をインストールする場合、Velero CSI プラグインを個別にインストールする必要はありません。した場合は、Velero ポッドの起動に失敗します。

たとえば、次のコマンドでは、AWS S3 ストレージ バックエンドとそれに対応する認証情報ファイルを使用して Velero をインストールします。Velero v1.14 であるため、Velero CSI プラグインを個別にインストールする必要はありません。

```
velero install \
  --provider aws \
  --plugins velero/velero-plugin-for-aws:v1.14 \
  --bucket velero-cpe-backup-bucket \
  --secret-file ./cloud-credential \
  --use-volume-snapshots=true \
  --features=EnableCSI --use-node-agent
```

以前のバージョンの Velero をインストールするには、Velero CSI プラグインもインストールする必要があります。例：

```
velero install \  
  --provider aws \  
  --plugins velero/velero-plugin-for-aws:v1.9.0,velero/velero-plugin-for-csi:v0.7.0 \  
  --bucket velero-cpe-backup-bucket \  
  --secret-file ./cloud-credential \  
  --use-volume-snapshots=true \  
  --features=EnableCSI --use-node-agent
```

# vSphere IaaS Control Plane での仮想マシン サービスの仮想マシンのバックアップとリストア

# 5

vSphere 管理者は、VMware vSphere Storage APIs – Data Protection (VADP) に基づくバックアップ パートナー ソリューションを使用して、スーパーバイザー で仮想マシン サービスの仮想マシンのバックアップ、フル リストア、および登録を自動的に実行します。基盤となるインフラストラクチャの問題が原因で自動登録が失敗した場合は、問題を修正してから、`registerVM` API を手動で呼び出して仮想マシンを再登録できます。

## 仮想マシン サービスの仮想マシンのバックアップ

vSphere IaaS Control Plane では、VMware vSphere Storage APIs – Data Protection を使用する Veeam などのバックアップ パートナー ソリューションを介して、仮想マシン サービスの仮想マシンを自動的にバックアップできます。

通常、vSphere 管理者はパートナー ソリューションを使用して次のタスクを実行します。

- バックアップ ソフトウェアのインストール、バックアップ ストレージとリポジトリの構成など、バックアップ インフラストラクチャを設定します。
- 仮想マシンまたは仮想マシンのグループを対象とするバックアップ ジョブを作成します。
- ジョブをトリガしてバックアップを開始します。

トリガされると、パートナー ソフトウェアは仮想マシンの構成、データ、および Kubernetes の状態をバックアップします。

通常、バックアップには次の項目が含まれます。

- vCenter Server に格納されている仮想マシンの構成。
- VMX ファイル。
- 静的または FCD の可能性がある仮想マシン データ ディスクの内容。

仮想マシン サービスの仮想マシンの場合、バックアップには仮想マシンの Kubernetes 状態と、リストア時に仮想マシンをブートストラップするために必要な追加リソースも含まれます。

VMware vSphere Storage APIs – Data Protection の詳細、およびパートナー バックアップ ソリューションの使用方法については、[ナレッジベースの記事 KB1021175](#) およびパートナーのドキュメントを参照してください。

## 仮想マシン サービスの仮想マシンのリストア

必要に応じて、vSphere 管理者はバックアップを使用して仮想マシンをリストアできます。たとえば、失敗した仮想マシンをリストアできます。

vSphere IaaS Control Plane は、仮想マシン全体をバックアップ ファイルから元の仮想マシンの最新状態にリストアする完全な仮想マシンのリストアのみをサポートします。

このタイプのリストアを実行するには、元の仮想マシンが スーパーバイザー と vSphere に存在しないことを確認します。まだ存在する場合は、リストア ジョブをトリガする前に、スーパーバイザー で `kubectl delete vm` コマンドを使用します。

バックアップ ソフトウェアからリストア プロセスをトリガすると、バックアップ ソフトウェアはリストア中に指定されたリソース プールおよびフォルダに仮想マシンを再作成します。仮想マシンは、同じ仮想マシン名または別の仮想マシン名にリストアできます。

プロセスが成功すると、vSphere IaaS Control Plane は、リストアされた仮想マシンを スーパーバイザー で検出し、最初に作成されたのと同じ vSphere 名前空間 に自動的に登録します。リストア プロセス中に、スーパーバイザー に VirtualMachine リソースが作成されます。必要に応じて、仮想マシンをブートストラップするためのシークレットや、仮想マシンで使用される追加ボリュームの PersistentVolumeClaims などの追加リソースも作成されます。

シークレットの詳細については、<https://kubernetes.io/docs/concepts/configuration/secret/> の Kubernetes ドキュメントを参照してください。パーシステント ボリュームの詳細については、<https://kubernetes.io/docs/concepts/storage/persistent-volumes/> を参照してください。

## ガイドラインと考慮事項

仮想マシンをリストアする場合は、次の点を考慮してください。

- リストア プロセスを開始する前に、次の手順を実行してください。
  - 元の仮想マシンが スーパーバイザー および vSphere から削除されていることを確認します。まだ存在する場合は、スーパーバイザー の `kubectl delete vm` コマンドを使用して削除します。
  - 基盤となるインフラストラクチャがバックアップとリストアの間で変更されていないことを確認します。元の仮想マシンが使用していたすべての適切なリソース（仮想マシン クラスやストレージ ポリシーなど）がターゲット vSphere 名前空間 にそのまま残っていることを確認します。  
そうしないと、仮想マシンの自動登録がエラーで失敗します。
- リストアされた仮想マシンは、最初に作成されたのと同じターゲット vSphere 名前空間 に登録されます。
- ターゲット リソース プールとフォルダ名は、元の仮想マシンののときと同じである必要があります。
- リストアされた仮想マシンの名前は、元の仮想マシン名と同じにすることができます。別の仮想マシン名を使用することもできます。

次のトピックを参照してください。

- [仮想マシン サービスの仮想マシンの手動登録](#)

## 仮想マシン サービスの仮想マシンの手動登録

何らかの理由で仮想マシンの自動登録が成功しない場合は、エラーメッセージが表示され、エラーの原因となっている正確な問題が通知されます。問題を修正したら、`registerVM` API を呼び出し、仮想マシンの `moID` を使用して仮想マシンを手動で登録できます。

次の例では、データセンター CLI (DCLI) コマンドを使用して仮想マシンを登録します。

### 手順

- 1 登録する仮想マシンの `moID` を取得します。

```
# dcli com vmware vcenter vm list
```

`moID` (MORef ID と呼ばれる) は、仮想マシン プリフィックスと、それに続く数値 ID で構成されます (vm-123456 など)。

- 2 仮想マシンを手動で登録します。

```
# dcli com vmware vcenter namespaces instances registervm --namespace my-namespace --vm vm-123456 +username my-username +password my-password
```

このコマンドは、次のようなタスクを返します。

```
task-637:6b051692-7aff-4d59-8a3f-699d114d37e3
```

- 3 VAPI タスク サービスを使用して、タスクのステータスを取得します。

```
# dcli com vmware cis tasks get --task task-637:6b051692-7aff-4d59-8a3f-699d114d37e3 +username my-username +password my-password
```

コマンドは、タスクのステータスを返し、失敗した場合はエラーメッセージを返します。

# Velero Plugin for vSphere を使用した vSphere ポッドのバックアップとリストア

## 6

Velero Plugin for vSphere を使用して、vSphere ポッド で実行されているワークロードをバックアップおよびリストアできます。

### 概要

Velero Plugin for vSphere を使用すると、スーパーバイザー 内の vSphere ポッド で実行されているワークロードをバックアップおよびリストアできます。vSphere ポッド で実行されているステートレス アプリケーションとステートフル アプリケーションの両方をバックアップおよびリストアできます。ステートフル アプリケーションの場合は、Velero Plugin for vSphere を使用して、パーシステント ボリューム (PV) のスナップショットを作成します。

**注：** スタンドアロンの Velero と Restic を使用して vSphere ポッド をバックアップおよびリストアすることはできません。スーパーバイザー にインストールされている Velero Plugin for vSphere を使用する必要があります。

### 前提条件

vSphere ポッド をバックアップおよびリストアする前に、Velero Plugin for vSphere をインストールして構成する必要があります。[#unique\\_20](#) を参照してください。

**注：** Velero Plugin for vSphere では、スーパーバイザー の状態のバックアップとリストアは行われません。

### vSphere ポッドのバックアップ

ステートレス vSphere ポッド をバックアップするには、次のコマンドを実行します。

```
velero backup create <backup name> --include-namespaces=my-namespace
```

バックアップは、すべてのローカル スナップショットが作成され、Kubernetes メタデータがオブジェクト ストアにアップロードされた後に Completed としてマークされます。ただし、ボリューム スナップショットのバックアップは非同期で実行されるため、バックグラウンドでまだ実行されている可能性があり、完了に時間がかかることがあります。

ボリューム スナップショットのステータスは、スナップショットおよびアップロード カスタム リソースを監視することにより確認できます。

## [スナップショット CRD]

スナップショット カスタム リソースは、スナップショットが作成された PVC と同じ名前空間に、ボリューム スナップショットごとに作成されます。PVC 名前空間のすべてのスナップショットを取得するには、次のコマンドを実行します。

```
kubectl get -n <pvc namespace> snapshot
```

スナップショット CRD には、`status.phase` フィールドのフェーズがいくつかあります。次に例を示します。

都道府県	説明
新規	まだ処理されていません
Snapshotted	ローカル スナップショットが作成されました
SnapshotFailed	ローカル スナップショットの作成に失敗しました
アップロードしています	スナップショットをアップロードしています
Uploaded	スナップショットがアップロードされました
UploadFailed	スナップショットのアップロードに失敗しました
Canceling	スナップショットのアップロードをキャンセルしています
キャンセルされました	スナップショットのアップロードがキャンセルされました
CleanupAfterUploadFailed	スナップショットのアップロード後のローカル スナップショットのクリーンアップに失敗しました

## [アップロード CRD]

アップロード CR は、オブジェクトストアにアップロードされるボリューム スナップショットごとに、Velero と同じ名前空間に作成されます。Velero 名前空間のすべてのアップロードを取得するには、次のコマンドを実行します。

```
kubectl get -n <velero namespace> upload
```

アップロード CRD には、`status.phase` フィールドのフェーズがいくつかあります。次に例を示します。

都道府県	説明
新規	まだ処理されていません
InProgress	アップロードが進行中です
UploadError	アップロードに失敗しました
CleanupFailed	アップロード後のローカル スナップショットの削除に失敗しました 再試行されます
Canceling	アップロードをキャンセルしています スナップショットのアップロードの進行中に <code>velero backup delete</code> が呼び出されると発生する場合があります
キャンセルされました	アップロードがキャンセルされました

[UploadError] では、アップロードが定期的に再試行されます。その時点でフェーズは [InProgress] に戻ります。アップロードが正常に完了すると、そのレコードは、一定期間保持され、最終的には削除されます。

## vSphere ポッド のリストア

Velero Plugin for vSphere を使用してバックアップされた vSphere ポッド ワークロードをリストアするには、次の手順を実行します。

- 1 リストアするワークロードの vSphere 名前空間 を作成します。
- 2 名前空間のストレージ ポリシーを構成します。
- 3 ワークロードをリストアするために、次の Velero コマンドを実行します。

```
velero restore create --from-backup backup-name
```

Velero リストアが Completed としてマークされるのは、ボリューム スナップショットと他の Kubernetes メタデータが現在のクラスタに正常にリストアされたときです。この時点で、このリストアに関連する vSphere プラグインのすべてのタスクも完了します。Velero バックアップの場合とは異なり、バックグラウンドの非同期のデータ移動タスクはありません。

Velero リストアが Completed になる前は、次のように CloneFromSnapshot/Download CR を監視することにより、ボリューム リストアのステータスを確認できます。

### [CloneFromSnapshots CRD]

各ボリューム スナップショットからリストアする場合、CloneFromSnapshot CR は、最初にスナップショットが作成された PVC と同じ名前空間に作成されます。PVC 名前空間のすべての CloneFromSnapshot を取得するには、次のコマンドを実行します。

```
kubectl -n <pvc namespace> get clonefromsnapshot
```

CloneFromSnapshot CRD には、status.phase フィールドのフェーズがいくつかあります。次に例を示します。

都道府県	説明
新規	スナップショットからのクローン作成が完了していません
完了	スナップショットからのクローン作成が完了しました
Failed	スナップショットからのクローン作成に失敗しました

### [ダウンロード CRD]

ダウンロード CR は、オブジェクト ストアからダウンロードされるボリューム スナップショットの各リストアから、Velero と同じ名前空間に作成されます。Velero 名前空間のすべてのダウンロードを取得するには、次のコマンドを実行します。

```
kubectl -n <velero namespace> get download
```

ダウンロード CRD には、status.phase フィールドのフェーズがいくつかあります。次に例を示します。

ステータス	説明
新規	まだ処理されていません
InProgress	ダウンロードが進行中です

ステータス	説明
完了	ダウンロードが完了しました
再試行	ダウンロードを再試行します。 バックアップ データのダウンロード中にエラーが発生した場合は、ダウンロードが再試行されます
Failed	ダウンロードに失敗しました

# vSphere IaaS Control Plane のバックアップとリストアのトラブルシューティング

# 7

vSphere IaaS Control Plane のバックアップとリストアに関連する問題をトラブルシューティングする方法について説明します。

次のトピックを参照してください。

- [バックアップからの スーパーバイザー のリストア後の実体のないオブジェクトのクリーンアップ](#)

## バックアップからの スーパーバイザー のリストア後の実体のないオブジェクトのクリーンアップ

バックアップから スーパーバイザー をリストアすると、バックアップの作成後に作成されるすべての Kubernetes リソースは、リストアの完了後に削除されます。仮想マシンやディスクなどのオブジェクトに関連付けられていたこれらのリソースの一部が、vCenter Server では実体なしになります。実体のないオブジェクトは vCenter Server からクリーンアップする必要があります。

## 手順

## 1 vSphere 名前空間 のすべての仮想マシンを一覧表示します。

- a スーパーバイザー 制御プレーン仮想マシンで次のコマンドを実行して、vSphere 名前空間 の folderMoId を取得します。

```

root@421c9fa40208448fecc15d277bdca66d [ ~ ]# kubectl get availabilityzone -o json
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "topology.tanzu.vmware.com/v1alpha1",
      "kind": "AvailabilityZone",
      "metadata": {
        ...
      },
      "spec": {
        "clusterComputeResourceMoIDs": [
          "domain-c50"
        ],
        "clusterComputeResourceMoId": "domain-c50",
        "namespaces": {
          "pod-ns": {
            "folderMoId": "group-v81", <--- [this] is the folderMoId that
you need for next step
            ...
          },
          "vmsvc-ns": {
            "folderMoId": "group-v83", <--- [this] is the folderMoId that
you need for next step
            ...
          }
        }
      }
    }
  ],
  "kind": "List",
  "metadata": {
    "resourceVersion": ""
  }
}

```

- b 次の DCLI コマンドを実行して、vSphere 名前空間 上の既存の仮想マシンをすべて一覧表示します。  
次の例では pod-ns 名前空間を使用します。

```

root@sc2-10-186-199-30 [ ~ ]# dcli +i +username 'Administrator@vsphere.local'
+password <password>
Welcome to VMware Datacenter CLI (DCLI)

usage: <namespaces> <command>

To auto-complete and browse DCLI namespaces: [TAB]

```

```

If you need more help for a command:          vcenter vm get --help
If you need more help for a namespace:       vcenter vm --help
To execute dcli internal command:            env
For detailed information on DCLI usage visit: http://vmware.com/go/dcli

dcli> com vmware vcenter vm list --folders group-v81
|-----|-----|-----|-----|-----|
|memory_size_MiB|vm   |name                                     |power_state|cpu_count|
|-----|-----|-----|-----|-----|
|512            |vm-84|deployment-before-backup-778449d88d-c9gnc|POWERED_ON |1        |
|512            |vm-85|deployment-before-backup-778449d88d-4jtgj |POWERED_ON |1        |
|512            |vm-86|deployment-before-backup-778449d88d-tqwbh |POWERED_ON |1        |
|512            |vm-91|deployment-after-backup-778449d88d-khkxx  |POWERED_OFF|1        |
|512            |vm-92|deployment-after-backup-778449d88d-7dgcc  |POWERED_OFF|1        |
|512            |vm-93|deployment-after-backup-778449d88d-sxbcf  |POWERED_OFF|1        |
|-----|-----|-----|-----|-----|

```

## 2 実体のない名前空間を見つけてクリーンアップします。

スーパーバイザー のバックアップの作成後に削除された名前空間は、スーパーバイザー のリストア後に Kubernetes リソースとして再作成されます。その Kubernetes リソースを削除する必要があります。

- a vCenter Server 内のすべての名前空間を一覧表示して、実体のない名前空間を見つけます。

```
dcli> com vmware vcenter namespaces instances list
```

- b すべての名前空間の Kubernetes リソースを一覧表示します。

```
root@423f9d75bef000dc828a535c6ac0bd4b [ ~ ]# k get ns -A
```

- a ステップ a とステップ b で見つけたオブジェクト間の差分を特定して、実体のない名前空間の Kubernetes リソースをクリーンアップします。

```
root@423f9d75bef000dc828a535c6ac0bd4b [ ~ ]# k delete ns test-set-workload-ns
namespace "test-set-workload-ns" deleted
```

### 3 VirtualMachine リソースに関連付けられている実体のない仮想マシンを見つけてクリーンアップします。

スーパーバイザー のバックアップの作成後に作成される VirtualMachine Kubernetes リソースは、スーパーバイザー がそのバックアップからリストアされると、実体のない仮想マシンになります。このような実体のない仮想マシンは、vCenter Server インベントリからクリーンアップする必要があります。

#### a VirtualMachine リソースに関連付けられている実体のない仮想マシンを見つけます。

次の手順では、vmsvc-ns 名前空間を例として使用します。

- 1 vCenter Server インベントリ内のすべての仮想マシンを一覧表示します。次の例（抜粋）では、group-96 が vmsvc-ns 名前空間に関連付けられているため、vCenter Server 内のすべての仮想マシンを一覧表示します。

```
dcli> com vmware vcenter vm list --folders group-v96
|-----|-----|-----|-----|-----|
|memory_size_MiB|vm   |name           |power_state|cpu_count|
|-----|-----|-----|-----|-----|
|2048           |vm-104|vmsvc-after    |POWERED_ON |2        |
|2048           |vm-97 |vmsvc-before   |POWERED_ON |2        |
|-----|-----|-----|-----|-----|
```

- 2 すべての VirtualMachine Kubernetes リソースを一覧表示します。kubectl get を実行して、いずれかの制御プレーン仮想マシンのリソースの詳細を取得し、uniqueID を出力で検索します。この例では、Kubernetes リソースに関連付けられている仮想マシン リストは vm-97 です。

```
root@42344b596f57bfcf9441179fa1eda5c [ ~ ]# k get vm -n vmsvc-ns -o json
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "vmoperator.vmware.com/v1alpha1",
      "kind": "VirtualMachine",
      ...
      "uniqueID": "vm-97",
      ...
    }
  ]
}
```

- 3 上記の手順で生成された 2 つのリストを比較します。

- vCenter Server 内の仮想マシン リスト：<vm-104, vm-97>
- Kubernetes リソースに関連付けられている仮想マシン リスト：<vm-97>

したがって、実体のない仮想マシン リストは <vm-104> です。

#### b 実体のない仮想マシンをクリーンアップします。

```
dcli> com vmware vcenter vm power stop --vm vm-104
dcli> com vmware vcenter vm delete --vm vm-104
```

4 ポッド リソースに関連付けられている実体のない仮想マシンを見つけてクリーンアップします。

スーパーバイザー のバックアップの作成後に作成されるポッドの Kubernetes リソースは、スーパーバイザー のリストア後に vCenter Server で実体のない仮想マシンになります。手順に従ってそれらを見つけてクリーンアップします。

次の例では名前空間 `pod-ns` を使用します。

- a vCenter Server インベントリ内のすべての仮想マシンを一覧表示します。

この例では、仮想マシン グループ `group-v83` が名前空間 `pod-ns` に関連付けられています。仮想マシン リストは、`vm-88`、`vm-89`、`vm-90`、`vm-101`、`vm-102`、`vm-103` です。

```
dcli> com vmware vcenter vm list --folders group-v83
|-----|-----|-----|-----|-----|
|
|memory_size_MiB|vm      |name                                     |power_state|
cpu_count|
|-----|-----|-----|-----|-----|
|
|512            |vm-101|deployment-after-backup-778449d88d-ldvn8 |POWERED_OFF|1
|
|512            |vm-102|deployment-after-backup-778449d88d-v29dd |POWERED_OFF|1
|
|512            |vm-103|deployment-after-backup-778449d88d-zdb19 |POWERED_OFF|1
|
|512            |vm-88  |deployment-before-backup-778449d88d-fgq5b|POWERED_ON  |1
|
|512            |vm-89  |deployment-before-backup-778449d88d-mp7td|POWERED_ON  |1
|
|512            |vm-90  |deployment-before-backup-778449d88d-cjqh6|POWERED_ON  |1
|
|-----|-----|-----|-----|-----|
|
```

- b Kubernetes リソースを一覧表示します。

`kubectl get` を実行していずれかの制御プレーン仮想マシンのリソースの詳細を取得し、`vmware-system-vm-moid` を出力で検索します。Kubernetes リソースに関連付けられている仮想マシン リストは `vm-88`、`vm-89`、`vm-90` です。

```
root@42344b596f57bfcf9441179faled1a5c [ ~ ]# k get pod -n pod-ns -o json
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "v1",
      "kind": "Pod",
      "metadata": {
        "annotations": {
          ...
          "vmware-system-vm-moid": "vm-90:5a5198fc-c5cb-4b89-
a70f-331025b40539",
          ...
        },
        ...
        "vmware-system-vm-moid": "vm-88:5a5198fc-c5cb-4b89-
a70f-331025b40539",
        ...
      }
    }
  ]
}
```

```
"vmware-system-vm-moid": "vm-89:5a5198fc-c5cb-4b89-  
a70f-331025b40539",  
...  
}
```

c 上記の手順で生成された 2 つのリストを比較します。

- vCenter Server 内の仮想マシン リスト: <vm-88, vm-89, vm-90, vm-101, vm-102, vm-103>
- Kubernetes リソースに関連付けられている仮想マシン リスト: <vm-88, vm-89, vm-90>  
したがって、実体のない仮想マシン リストは <vm-101, vm-102, vm-103> です。

d 実体のない仮想マシンをクリーンアップします。

```
dcli> com vmware vcenter vm delete --vm vm-101  
dcli> com vmware vcenter vm delete --vm vm-102  
dcli> com vmware vcenter vm delete --vm vm-103
```

- 5 Tanzu Kubernetes Grid クラスタに関連付けられている実体のない仮想マシンとリソース プールを見つけてクリーンアップします。

スーパーバイザー のバックアップの作成後に作成される Tanzu Kubernetes Grid クラスタも、スーパーバイザー のリストア後に vCenter Server で実体のない仮想マシンになります。

**注：** リストア操作後、TKG クラスタの作成中に問題が発生した場合は、現在の手順の指示に従って実体のない仮想マシンをクリーンアップする必要があります。

- a 実体のない Tanzu Kubernetes Grid クラスタ リストを見つけてみます。

kubectl を使用して、いずれかの制御プレーン仮想マシンの Tanzu Kubernetes Grid クラスタの Kubernetes リソース リスト (<test-cluster, test-cluster-e2e-script, tkc-before-backup>) を取得します。

```
root@4239f4159c7063d5608cf3fc0bdd532e [ ~ ]# k get tkc -A
NAMESPACE          NAME          CONTROL PLANE  WORKER  TKR
NAME               AGE  READY  TKR COMPATIBLE  UPDATES  AVAILABLE
selfservice-tkc-ns test-cluster          1          1      v1.23.8---
vmware.3-tkg.1    19h  True   True
test-gc-e2e-demo-ns test-cluster-e2e-script 3          1      v1.23.8---
vmware.3-tkg.1    18h  False  True
tkc-ns            tkc-before-backup    3          1      v1.23.8---
vmware.3-tkg.1    16h  True   True
```

DCLI を使用して、名前空間または Tanzu Kubernetes Grid クラスタに関連付けられているすべてのリソース プールを取得し、vCenter Server 内の Tanzu Kubernetes Grid クラスタ リスト (<test-cluster, test-cluster-e2e-script, tkc-before-backup, tkc-after-backup>) を取得します。

```
dcli> com vmware vcenter resourcepool list
|-----|-----|
|name          |resource_pool|
|-----|-----|
|Resources     |resgroup-10  |
|Resources     |resgroup-23  |
|Namespaces    |resgroup-56  |
|selfservice-tkc-ns |resgroup-62  | <--- this is a namespace
|test-cluster   |resgroup-66  | <--- Tanzu Kubernetes Grid cluster
|test-gc-e2e-demo-ns |resgroup-70  | <--- this is a namespace
|test-cluster-e2e-script|resgroup-74  | <--- Tanzu Kubernetes Grid cluster
|tkc-ns        |resgroup-80  | <--- this is a namespace
|tkc-before-backup |resgroup-89  | <--- Tanzu Kubernetes Grid cluster
|tkc-after-backup |resgroup-96  | <--- Tanzu Kubernetes Grid cluster
|-----|-----|
```

- b 上記の手順で生成された 2 つのリストを比較します。実体のない Tanzu Kubernetes Grid クラスタ リストは <tkc-after-backup> です。

- c 実体のない Tanzu Kubernetes Grid クラスタに関連付けられている仮想マシンをクリーンアップします。

DCLI を使用し、関連付けられているリソース プール <resgroup-96> を使用して、実体のない Tanzu Kubernetes Grid クラスタに関連付けられているすべての仮想マシンを取得します。

```
dcli> com vmware vcenter vm list --resource-pools resgroup-96
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|
|memory_size_MiB|vm      |name                                     |power_state|
|cpu_count|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|
|2048           |vm-100|tkc-after-backup-zlcdm-wk5xf           |POWERED_ON |
|2             |      |                                         |            |
|2048           |vm-101|tkc-after-backup-zlcdm-76q4h           |POWERED_ON |
|2             |      |                                         |            |
|2048           |vm-98 |tkc-after-backup-zlcdm-9fv2w           |POWERED_ON |
|2             |      |                                         |            |
|2048           |vm-99 |tkc-after-backup-workers-4hdqb-657fb58d45-d7pqq|POWERED_ON |
|2             |      |                                         |            |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|
```

次に、仮想マシンを 1 台ずつ削除します。

```
dcli> com vmware vcenter vm power stop --vm vm-100
dcli> com vmware vcenter vm delete --vm vm-100
```

- d 実体のない Tanzu Kubernetes Grid クラスタに関連付けられているリソース プールをクリーンアップします。

```
<dcli> com vmware vcenter resourcepool delete --resource-pool resgroup-96
```

実体のないリソース プールを vSphere Client から削除することもできます。

- 6 永続ボリューム (PV) に関連付けられている実体のない FCD (First Class Disk) を見つけてクリーンアップします。

スーパーバイザー のバックアップの作成後に作成される PV の Kubernetes リソースは、スーパーバイザーのリストア後に vCenter Server で実体のない FCD になります。手順に従ってそれらを見つけてクリーンアップします。

- a PV に関連付けられた実体のない FCD を見つけます。

- 1 実体のない FCD を見つけるために使用する govc をインストールします。govc は、ユーザー インターフェイスに代わる使いやすい CLI で、自動化タスクに適しています。

```
curl -L -o - "https://github.com/vmware/govmomi/releases/latest/download/govc_$(uname -s)_$(uname -m).tar.gz" | tar -C /usr/local/bin -xvzf - govc
```

その他のインストール オプションについては、<https://github.com/vmware/govmomi/tree/main/govc#installation> を参照してください。

- 2 次の bash スクリプトを実行して、スーパーバイザー に存在する PV を一覧表示します。

```
#!/bin/bash

export GOVC_INSECURE=1
export GOVC_USERNAME='Administrator@vsphere.local'
export GOVC_PASSWORD=<password>
export GOVC_URL=https://<vc ip>/sdk

# datastore path example - /test-vpx-1688432886-30489-wcp.wcp-sanity/datastore/
sharedVmf-0
govc volume.ls -l -ds=<datastore path>
```

結果は次のとおりです。

```
peiyangs@peiyangs-a01 govc % sudo bash orphanedPV.sh
590c8e31-f5bf-4179-9250-5cdd66bf591c
pvc-843c932b-8974-475d-8f8a-9b165137169d    1.0GB    KUBERNETES
vSphereSupervisorID-7f88d7b3-12ac-4fcf-a101-b80eb76becdf
37f8ad5b-dfe6-465b-b0f0-11591a2968dc    pvc-77c42590-
f0b0-457f-9743-6a3ebca55078    1.0GB    KUBERNETES
vSphereSupervisorID-7f88d7b3-12ac-4fcf-a101-b80eb76becdf
28a265b8-2e6b-421c-b16d-046ffc7aeea7    pvc-1b88c923-4354-4537-a7cb-
a8a6d763d5e7    1.0GB    KUBERNETES    vSphereSupervisorID-7f88d7b3-12ac-4fcf-a101-
b80eb76becdf
```

- 3 次の bash スクリプトを実行して、vCenter Server 内のすべてのディスクを一覧表示します。

```
#!/bin/bash

export GOVC_INSECURE=1
export GOVC_USERNAME='Administrator@vsphere.local'
export GOVC_PASSWORD=<password>
export GOVC_URL=https://<vc ip>/sdk
```

```
# datastore path example - /test-vpx-1688432886-30489-wcp.wcp-sanity/datastore/
sharedVmfs-0
govc disk.ls -l -ds=<datastore path>
```

結果は次のとおりです。

```
peiyangs@peiyangs-a01 govc % sudo bash orphanedPV.sh
28a265b8-2e6b-421c-b16d-046ffc7aeea7 pvc-1b88c923-4354-4537-a7cb-a8a6d763d5e7
1.0G Jul 4 02:33:27 <--- this is the disk correspondings to PV
37f8ad5b-dfe6-465b-b0f0-11591a2968dc pvc-77c42590-f0b0-457f-9743-6a3ebca55078
1.0G Jul 4 02:32:41 <--- this is the disk correspondings to PV
3a7517c2-f8c2-46a9-b0d5-18c665759311 vmware-sv-img-cache-domain-c50
26.0M Jul 4 02:36:41
590c8e31-f5bf-4179-9250-5cdd66bf591c pvc-843c932b-8974-475d-8f8a-9b165137169d
1.0G Jul 4 02:30:45 <--- this is the disk correspondings to PV
68ba220c-0f83-49eb-b77a-d60471e24844 pvc-92f83ae0-7c2d-46d9-ab85-19858462ddd1
5.0G Jul 4 18:27:02 <--- this is the disk correspondings to PV
72dbe8c5-a3b5-4298-8203-ealcb86116e6 vmware-sv-img-cache-domain-c50
3.0M Jul 4 02:38:39
79e233a6-0134-40e7-8ba8-3133442324f9 vmware-sv-img-cache-domain-c50
195.0M Jul 4 18:26:12
a1a0a9d7-0baf-4592-9041-8c0feb960246 vmware-sv-img-cache-domain-c50
7.0M Jul 4 02:35:37
cec2af09-80af-4086-a069-34140e2480dc vmware-sv-img-cache-domain-c50
193.0M Jul 4 02:31:12
```

4 上記の手順で生成された 2 つのリストを比較します。

- PV のリスト : <590c8e31-f5bf-4179-9250-5cdd66bf591c, 37f8ad5b-dfe6-465b-b0f0-11591a2968dc, 28a265b8-2e6b-421c-b16d-046ffc7aeea7>
- FCD のリスト : <590c8e31-f5bf-4179-9250-5cdd66bf591c, 37f8ad5b-dfe6-465b-b0f0-11591a2968dc, 28a265b8-2e6b-421c-b16d-046ffc7aeea7, 68ba220c-0f83-49eb-b77a-d60471e24844>

実体のない FCD は <68ba220c-0f83-49eb-b77a-d60471e24844> です。

b 実体のない FCD を削除します。

govc を使用して実体のない FCD を削除します。スクリプトの例は次のとおりです。

```
#!/bin/bash

export GOVC_INSECURE=1
export GOVC_USERNAME='Administrator@vsphere.local'
export GOVC_PASSWORD=<password>
export GOVC_URL=https://<vc ip>/sdk

# datastore path example - /test-vpx-1688432886-30489-wcp.wcp-sanity/datastore/
sharedVmfs-0
govc disk.rm -ds=<datastore path> 68ba220c-0f83-49eb-b77a-d60471e24844
```

結果は次のとおりです。

```
peiyangs@peiyangs-a01 govc % sudo bash orphanedPV.sh  
[06-07-23 11:36:27] Deleting 68ba220c-0f83-49eb-b77a-d60471e24844...OK
```